Industrial Data Communications

4th Edition

By Lawrence M. Thompson

ISA

Setting the Standard for Automation™

Industrial Data Communications

4th Edition

By Lawrence “Larry” M. Thompson

Notice

The information presented in this publication is for the general education of the reader. Because neither the author nor the publisher have any control over the use of the information by the reader, both the author and the publisher disclaim any and all liability of any kind arising out of such use. The reader is expected to exercise sound professional judgment in using any of the information presented in a particular application.

Additionally, neither the author nor the publisher have investigated or considered the affect of any patents on the ability of the reader to use any of the information in a particular application. The reader is responsible for reviewing any possible patents that may affect any particular use of the information presented.

Any references to commercial products in the work are cited as examples only. Neither the author nor the publisher endorses any referenced commercial product. Any trademarks or trade names referenced belong to the respective owner of the mark or name. Neither the author nor the publisher makes any representation regarding the availability of any referenced commercial product at any time. The manufacturer's instructions on use of any commercial product must be followed at all times, even if in conflict with the information in this publication.

Printed in the United States of America.
10 9 8 7 6 5 4 3 2

ISBN 978-1-934394-24-3 Ebook ISBN 978-1-937560-59-1 PDF ISBN 978-1-937560-88-1

ISA
67 Alexander Drive
P.O. Box 12277
Research Triangle Park, NC 27709

Library of Congress Cataloging-in-Publication Data

Thompson, Lawrence M.
Industrial data communications / By Lawrence "Larry" M. Thompson. --
4th ed.
p. cm.
ISBN-13: 978-1-934394-24-3 (pbk.)
1. Data transmission systems. I. Title.
TK5105.T46 1997
004.6--dc22
2007036855

ISA Resources for Measurement and Control Series (RMC)

- *Control System Documentation: Applying Symbols and Identification, 2nd Edition*
- *Control System Safety Evaluation and Reliability, 2nd Edition*
- *Industrial Data Communications, 4th Edition*
- *Industrial Flow Measurement, 3rd Edition*
- *Industrial Level, Pressure, and Density Measurement, 2nd Edition*
- *Measurement and Control Basics, 4th Edition*
- *Programmable Controllers, 4th Edition*

This Book Is Dedicated To

The many practitioners of various disciplines who, through no fault of their own, have arrived at the position of needing knowledge of industrial data communications just to survive.

Contents

Preface to Fourth Edition

Rationale

In the fifteen or so years since the first edition of this book, nearly every aspect of data communications has changed, and above all industrial applications. The original rationale for this book was that many people are forced to learn data communications because the processes aren't as transparent and as "plug and play" as they should be. Though these individuals never intended to become experts in data communications, they are nonetheless now forced to learn some specific detailed facts just to accomplish their primary job functions.

Unfortunately, fifteen years later, there is still a need to understand the technical jargon and polemics of data communications. Though you will not find it difficult or even tedious to acquire the necessary knowledge of data communications, the material must be organized in a way that helps you stay focused on the key points. This edition provides that framework while also containing significant new material to encompass the changes in technology and indeed in the direction and focus of industrial applications since the third edition. Specifically, I have expanded coverage of the different fieldbuses, of industrial Ethernet and wireless technologies, and of the security considerations that have become ubiquitous in industrial use.

The need to upgrade the third edition became apparent much as it had with the second edition: as soon as it was published. The field of data communication is quite dynamic. Though the fundamentals have not changed (or changed very little) industrial applications are changing at a quicker pace, and unfortunately, much quicker than the revision cycles of the texts that hope to cover them. Though much of the previous three editions are still quite valid, this edition required more than a minor revision, and considerable freshening was needed to ensure that the materials are not dated.

Objectives

The objectives of the fourth edition are exactly the same as those of the previous ones: to introduce the principles and applications of industrial data communications and bring you to a level where you can communicate with other professionals on this topic. Because of the changes in this field, particularly since the third version, this book assumes you are familiar with Internet use and perhaps some data communications applications. It is written in the same conversational style as its predecessors, with the hope that this informality maximizes your understanding.

Audience

The intended audience is the person with some general technical education who is somewhat literate with computers. Though knowledge of the electrical-electronic disciplines will aid understanding, it is still not a prerequisite, as quick (and simplistic) explanations of the concepts necessary for understanding are given in the appendices. However, as in previous editions, a willingness to understand new concepts and a sense of historical perspective will help. As always when reading this text, patience as well as a sense of humor will be found high on the list of requirements.

Topics

As in all previous versions, the text ranges from simple basics to the complex applications. A familiarity with basic number systems along with hexadecimal representation is required; though, here again, these are explained to the necessary level of complexity in the appendices. As in the third edition, this edition's appendices also cover modulation and analog/digital conversion fundamentals. For more detail on these subjects, many, many, reference books, study materials, and computer-assisted training courses are available. This book is not a design or engineering document but a primer designed to bring your knowledge quickly up to the current practice. It is not assumed that you already know, or are even familiar with, the subject of this text, so we discuss concepts and applications only in the detail needed to grasp general concepts and/or applications.

Larry Thompson
Owner/General Manager
ESdatCo
707 Coleman St.
Marlin, TX 76661
larrymthompson@hotmail.com

Acknowledgments

Whenever a book of this nature and length is created, there are many persons who contribute greatly besides the author. I cannot name them all; however, here are a few of the persons responsible for this book being as successful as it is today:

Susan Colwell, ISA Manager of Publications Development, who managed to take my mangled text and graphics and place them in a coherent order;

Tim Shaw, Technical Reviewer, who contributed greatly to the technical accuracy of this volume and whose comments and guidance were greatly appreciated;

To the many users of this book who have contributed ideas, wishes, and technical comments, and to whom this book was originally, and still is, dedicated;

And lastly, to my wife, Gavina, who gave up her time with me that this book could be developed, corrected, and finally brought to fruition.

Larry Thompson

About the Author

Larry Thompson has been an ISA adjunct instructor since 1984 and has designed, developed, taught, and maintained industrial controls and networks in many varied applications. After earning his Bachelor of Applied Sciences from Tarleton State University, he began a career in data communications, including experience as a test engineer and test engineering supervisor for Reliance Telecom and as an instructor in instrumentation, computer networking/system administration, and e-commerce technology for Texas State Technical College.

Larry also served twenty years in the U.S. Air Force, primarily in electronic encryption systems. In 1979 he started Electronic Systems development and training company (ESdatCo), a technical services business, and is presently under contract as an IT technical manager for a financial institution with eight remote sites. Larry is a Certified Control Systems Technician and presently instructs in ISA's Certified Automation Profession exam review course. He holds an FCC Radiotelephone License (General, formerly First Class).

Larry is the author of the following ISA books: *Industrial Data Communications* (4th edition) and *Basic Electricity and Electronics for Control: Fundamentals and Applications* (3rd edition).

1
Communication Concepts

This chapter deals with the fundamentals of data communications. We primarily probe and discuss the factors that affect all communications from a "big picture" perspective, so that when the details are presented in later chapters, you will know which niche is being filled in. A portion of the chapter—the discussion of low-level data organization sometimes called a "coding" (referring to ASCII and EBCDIC)—is almost a technical history in itself (more on this subject is provided in appendix B).

Goals

Communication has certain goals. In all communications there must be a source (typically called the transmitter) and one or more destinations (typically called receivers). The goal is to go from one end user (the source) to the other end user or users (the destination). This transmitting is done through a medium of one kind or another, varying according to the technology used. Audio communications use the air as a medium. In data communications we use conductors (usually referred to as "copper" connections), light pipes (referred to as fiber-optic cabling), and electromagnetic transmission (usually referred to as wireless or radio).

Basically, this book is all about how we *organize* and move data. Data itself is useless until organized, at which time it becomes *information*. We organize communication processes in three basic ways: point-to-point, multi-drop, or networked. Figure 1-1 illustrates these three organizations. Note that while these terms are couched in technical symbology, the concepts behind them are relatively easy. Point-to-point means just that, from one point to another, or from one end user to another directly. Multi-drop is closer to a network than to point-to-point. Generally, multi-drop involves a master of some kind with slave stations, not peers. It should be understood that by most definitions of a network, a multi-drop system is a network. Finally, a network is simply three or more stations connected by a common media, by which they may share information. Later in this book we will tighten our definition, dividing networks into wide area or local, and so on. But for now, where three or more stations are connected we will consider them a network.

Figure 1-2 illustrates the three different categories of communications: simplex, half-duplex, and duplex. These terms (i.e., simplex, half-duplex, and duplex) may be used to describe all types of circuitry or modes of transmission, whether they are point-to-point, multi-dropped, or networked. It is important that you understand these three terms because almost all descriptive language pertaining to data communications uses them.

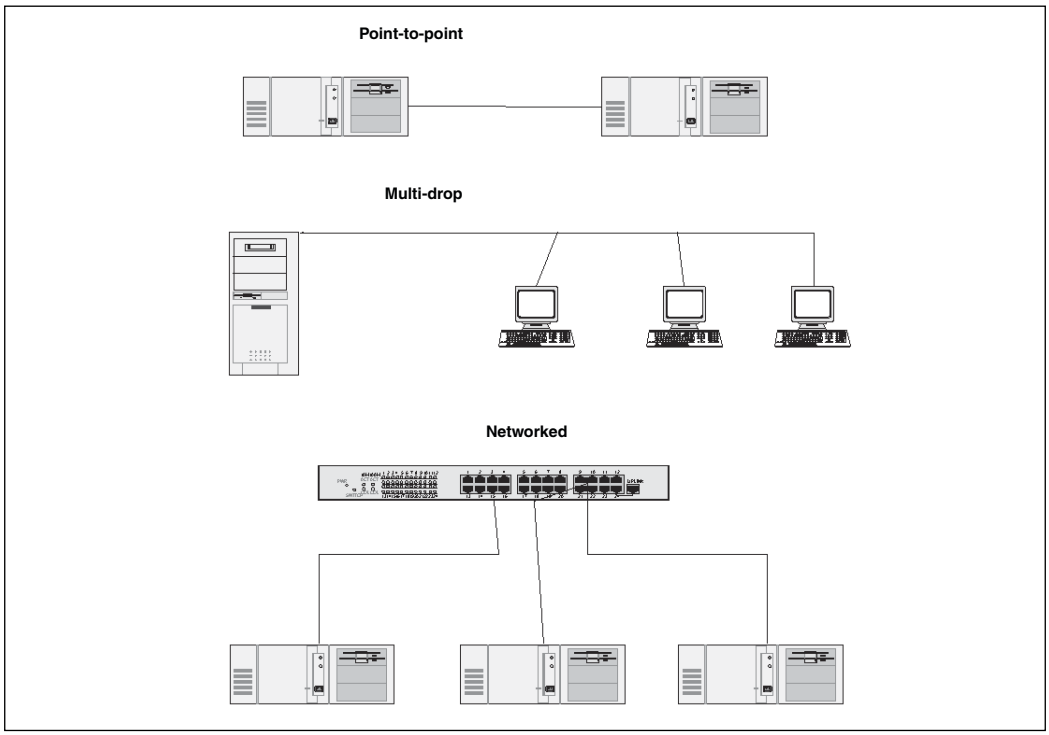


Figure 1-1. Communications Organization

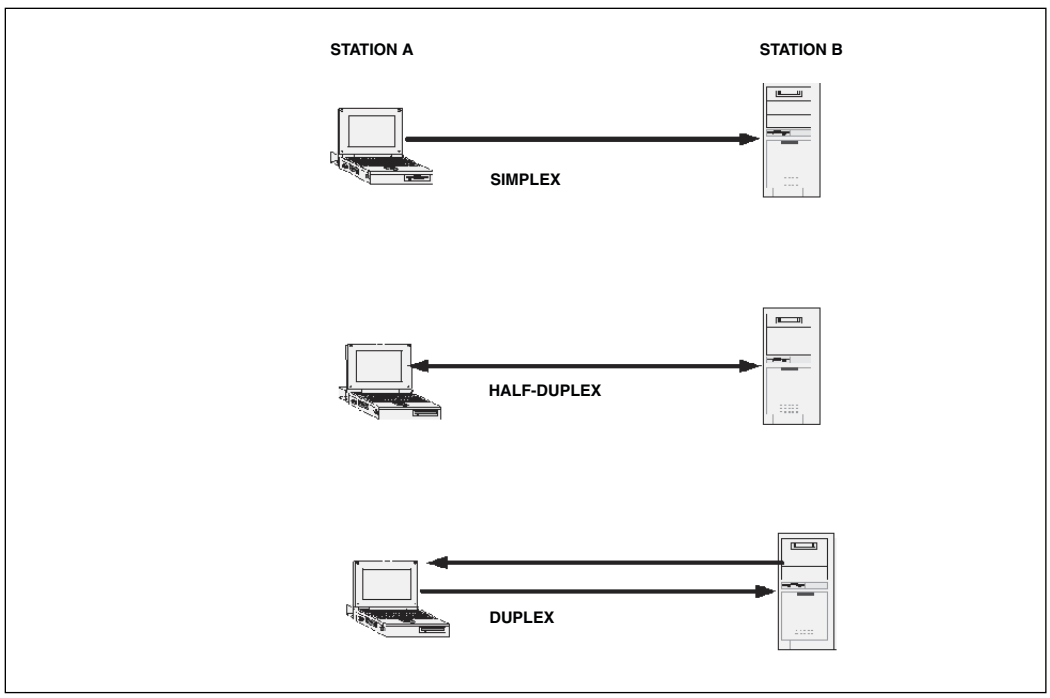


Figure 1-2. Modes and Circuits

NOTE: The differences between a "mode" and a "circuit" are rather arbitrary.

A circuit is typically the actual communications channel hardware configuration. Mode can refer to either the hardware configuration or a "virtual" circuit, which is a channel that consists of software or other processes that are not directly connected but are communicating entities. Hence, a virtual circuit refers more to the process of communications than to an actual hardware configuration.

The reader should be aware that these constraints may be due to hardware or software. If the hardware is duplex, the software may make it run half-duplex, however, if the hardware will not support a mode of operation, no amount of software will cause it to actually be so, although in the virtual world it may appear to be so to us, an example would be a half-duplex system that appears (due to the speed and message constraints) to be duplex to the user.

The three communications categories are as follows:

Simplex or Unidirectional Mode. In this mode communication occurs only in one direction, never in the opposite direction; in this case it is from A to B. The circuit that provided this operation was originally called simplex, but this leads to confusion with telephony terminology. Unidirectional is the name of the mode of transmission, and using it for this circuit would be much more descriptive.

Half-Duplex Mode. In this mode, communication may travel in either direction, from A to B or B to A but not at the same time. Half-duplex functions much like human conversation does, that is, one speaker at a time and one (or more) listener(s).

Duplex Mode. In duplex (the circuit is still referred to as "full duplex"), communication can travel both directions simultaneously: A to B and B to A at the same time.

Serial and Parallel Transmission

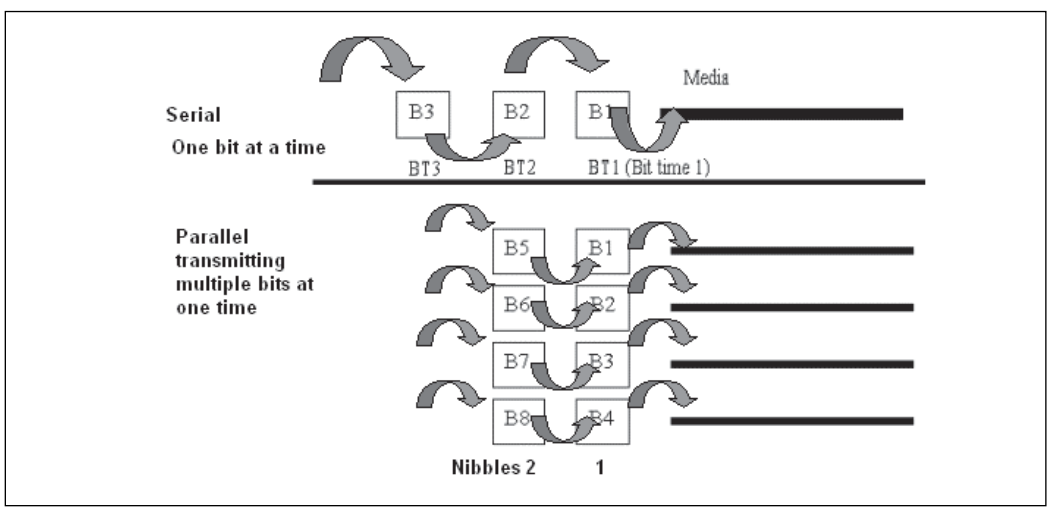


Figure 1-3. Serial and Parallel Concepts

Serial transmission (see figure 1-3) has one channel (one medium of transmission), and every bit follows one after the other, much like a group of people marching in single file. This means the expense of only one channel is required to send bits at much higher speed in order to achieve the same throughput as parallel transmission.

Parallel transmission is where the signal must traverse more than one transmission channel, as in a group of people marching four (or more) abreast. For a given message, a two-channel parallel will transmit twice as much information as a serial channel running at the same line speed. However, a parallel transmission running any appreciable distances will entail two serious problems. First, the logistics of having two (or more) parallel media is sure to double (or more) your costs. Second, assuring their simultaneous reception over some distance is technically quite difficult, along with ensuring that cross-talk (a signal from one transmission line being coupled on to another) is kept low. Cross-talk increases with signaling rate, so trying to go faster with multiple conductors becomes increasingly difficult. Figure 1-4 illustrates what the signals would look like in parallel transmission. Note that for the two 4-bit combinations it took two timing periods to transmit all eight, whereas the serial signal took eight timing periods. The catch is that the serial signal only needed one media channel, while the parallel signal needed four media channels.

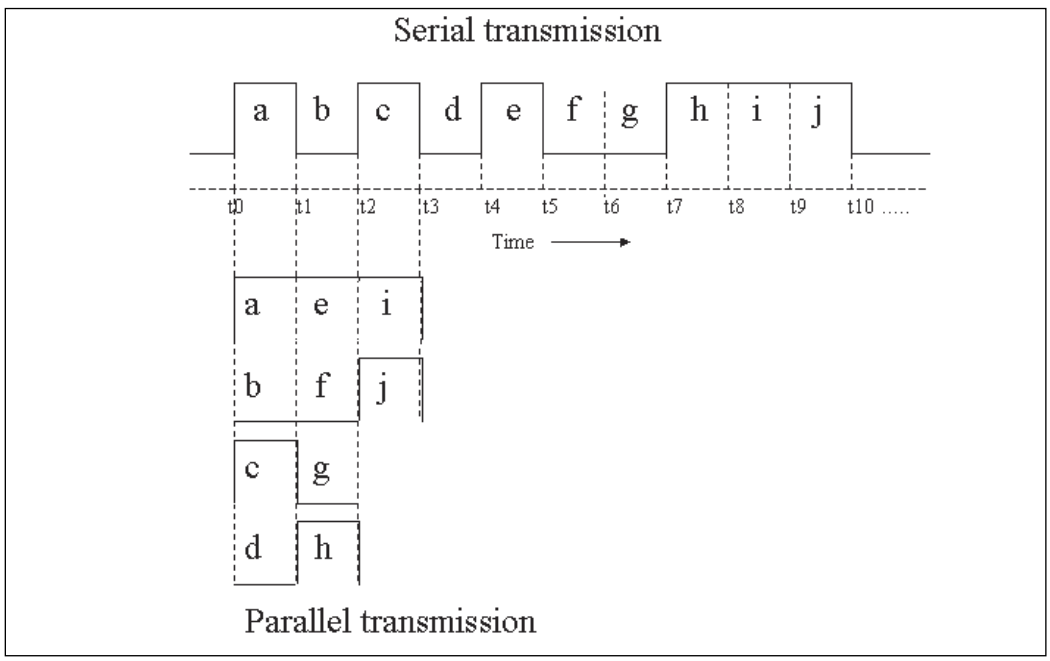


Figure 1-4. Serial and Parallel Signals

For these reasons, most of our transmissions outside of the computer cabinet at any but very low speeds will be by serial transmission. Inside the computer cabinet (and indeed up to about 1 meter outside it) the buses have in the past used parallel transmission since the necessity of high speed outweighs the cost. However, as bus speeds continue to increase,

newer technologies (PCIe, SATA) use serial transmission. This is because the problems of maintaining transition synchrony over the parallel bus increase drastically as speed increases. Over the past two decades PC printers have been parallel (they started out as serial) because the signal bit lengths are long enough in duration (the signal is slow enough) to permit parallel transmission over a limited distance. These are being replaced by USB and other technological advances in serial transmission.

Data Organization: Signals

Digital Signals

With binary digital signals we have two states for our data: a binary digit is either a "one" or a "zero." Do not confuse "zero" with nothing. Zero conveys half the information in a binary signal—think of it more as true or false, with no allowance for maybe. Digital signals other than binary exist. The definition of a digital signal is one that has discrete states, which could be two (binary), three (trinary), or even ten (decimal). A binary digit (the contraction of this term is "bit") will be either in one or another state, either true or false, yes or no, up or down, left or right, on or off, one or zero.

Now that we have defined channels, data, and bits, what are we going to send as a signal over our channel? Usually, it is a pattern of bits. Bits alone are just data. We must organize our data into some form so it becomes information. When this is done at higher levels of organization we may call this organization "protocols" or even "application programming interfaces." At their very lowest level, a primitive level you might say, the patterns representing data are called codings.

A coding is properly a "shorthand" representation of a signal. It is a generally understood representation. Codings are not used for secrecy—ciphers are. A "standard signal" could be called a coding and in fact are referred to as such in the daily work of industry. A standard signal is one that has the approval of the users and/or a standardization agency and is a specified way to organize data. This text focuses entirely on how data is organized for different functional tasks. There are many different organizations of digital signals: standards (approved organizations), open (in general use), and proprietary. In this book we will only touch upon a few.

Analog Standard Signals

In all areas of data communications, a number of "standard signals" exist. An analog (the word means model) signal is one that may be at any value between its upper and lower limits. This contrasts with a binary digital signal, which only has one of two values. Perhaps the easiest of these standard signals to visualize would be the standard 4-20 mA current loop signal used in process measurement and control (see figure 1-5).

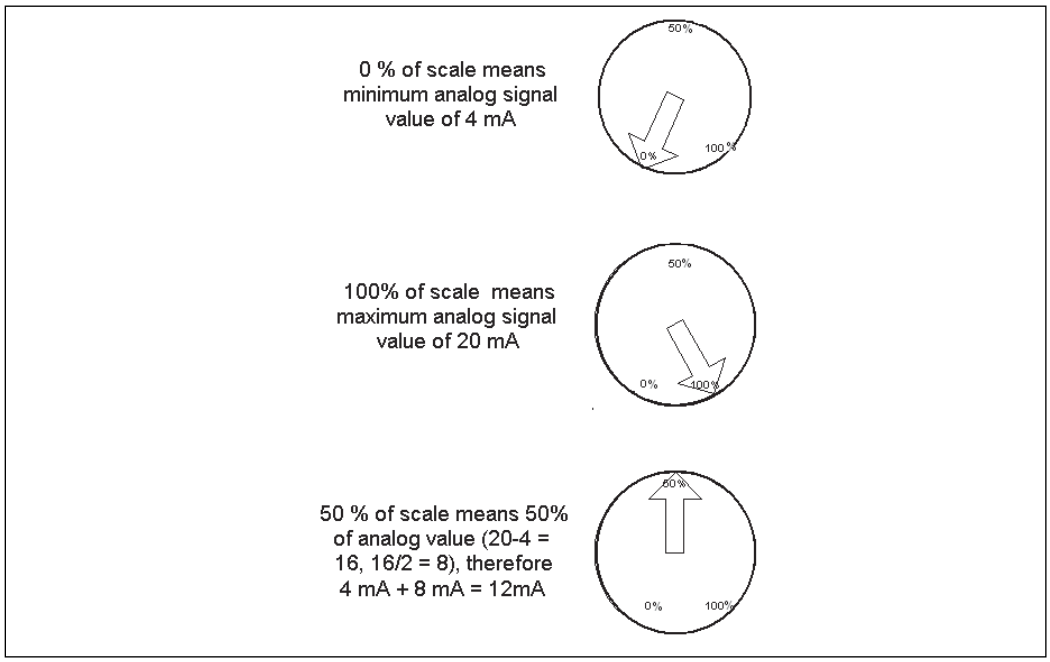


Figure 1-5. Process Analog Signals

These signals represent 0% to 100% of the range (full scale) specified. All instrumentation readings could be likened to a meter face in which any value is allowed between 0 and 100%, perhaps specified in engineering value measurements. An example would be 200° to 500°C. The 4-to-20-milliamp (mA) standard instrument signal would represent 200° as 4 mA and 500° as 20 mA. These are electrical values for the quantity of energy used in signaling. If you are not familiar with these units you may not fully understand some of the standards and their implications. This text will try to draw the conclusions for you. However, any basic electrical text (particularly one for non-technical people) will provide more than enough background to help you understand the terms used in this book.

The 4-to-20-mA signal is the most commonly used data communications standard and is the one used in the two-wire loop. The ISA SP50 committee selected a current loop because it is low impedance, has a greater immunity to noise than a high-impedance voltage circuit, and it can power the loop instrument. The receiving devices themselves are high impedance (voltage input) and acquire their input across a 250-ohm resistor (1-5V). This allows a known loading for each receiver in the loop. Most contemporary devices use 1 to 5 volts DC. Conversion between current and voltage is provided by a 250-ohm resistor in the current loop. This simple arrangement allows the two-wire loop to both power the loop instrument and provide the measurement signal. Open (nonproprietary) standards allow users to pick and choose among vendors with the confidence that the inputs and outputs are compatible. The standard signals also allow a vendor to manufacture for a larger group of users than if there were only proprietary signals.

Digital Standard Signals

There have been and still are standard digital signals in the telephone and digital data communications area (we will cover these as we go along). However, in the measurement and control areas no open, all-digital standard signal has been used as an alternative to the signal described in the last section. The international standard for an industrial fieldbus (IEC 61158) defines eight different fieldbuses, four of which—including Profibus PA and Foundation Fieldbus—are primarily used in process control. Though all these fieldbuses are set as standards they are not necessarily compatible with each other. It will take the marketplace to determine which one will actually be dominant and become the de facto standard.

Process measurement and control does use several digital communications standards—such as EIA/TIA 232(F) or EIA/TIA 485(A)—that actually place the data on the media. We discuss these in the chapter on Serial Communications. The Electronic Industries Alliance (formerly Electronic Association, formerly Radio & Television Manufacturers Association formerly… all the way back to 1924) is an association of manufacturers that develops standards. TIA (Telecommunications Industry Association) is a subdivision of EIA.

Data Organization: Communications Codes

Communication codes are the lowest level of data organization and are designed for interface with humans. These codes are the representations of letters, numerals, and control codes that are stored and recalled, printed, sorted, and in general processed. IBM™ designed one of the first digital communications codes. The IBM™ 4 of 8 Code allowed error detection on a character-by-character basis. This section explains this and other communications codes.

The IBM 4 of 8 Code was a proprietary code. Typically, other manufacturers had their own proprietary codes, all of which were incompatible with each other. In 1961 the U.S. government released the American Standard Code for Information Interchange (ASCII). Several months later IBM released its own code: EBCDIC, for "Extended Binary Coded Decimal Interchange Code." Because the U.S. government was (and is) a large buyer of data equipment, ASCII gradually gained in acceptance with many vendors (other than IBM), who relinquished their proprietary codings and adopting ASCII.

The IBM 4 of 8 Code

The IBM 4 of 8 Code is illustrated in table 1-1. In it, for any particular character there will be four ones and four zeros, that is, 8 bits, or an octet. This arrangement was used to detect character errors, but it carried a large overhead; that is, the ratio of elements used for error detection to those for data transmission is high. About 70 characters were used out of the 256 available.

Table 1-1. IBM 4 of 8 Coding

Character	Bit Arrangement	Character	Bit Arrangement
0	01011001	I	10010110
1	10000111	J	10001011
2	01000111	K	01001011
3	11001001	L	11000011
4	00100111	M	00101011
5	10101001	N	10100011
6	01101001	O	01100011
7	11101000	P	11100010
8	00010111	Q	00011011
9	10011001	R	10010011
A	10001110	S	01001101
B	01001110	T	11000101
C	11000110	U	00101101
D	00101110	V	10100101
E	10100110	W	01100101
F	01100110	X	11100100
G	11100001	Y	00011101
H	00011110	Z	10010101

Example 1-1:
Problem: The following signal is received. Decode the signal (using table 1-1) and determine which character has an error.

The transmission scheme uses 1 start bit and 1 stop bit. In order to tell when a transmission line is idle, a continuous one was placed on the line. To indicate a start the first bit was always a zero. Nine bits (start bit plus the 8 bits) later, the signal placed a 1 bit on the line to indicate it had stopped transmitting a character and remained in that condition until the next start bit.

000011110101010011010110000111011000111100110000111

Solution: First break the code up into 10-bit groups (one start bit, one stop bit, and eight information bits):

0000111101 0101001101 0110000111 0110001111 0011000111

Second, strip off the start and stop bits:

00011110 10100110 11000011 11000111 01100011

Third, convert the bits into alpha characters:

H E L ? O

The second L has 5 ones and 3 zeros; it is a character error.

International Telegraphic Alphabet (ITA) #5

Also known as the American Standard Code for Information Interchange, or ASCII for short, the International Telegraphic Alphabet (ITA) #5 (see table 1-2) does not make use of letter frequencies. It does use a numerically arranged code in which the alphabet is in ascending order, which permits sorting operations and the like. ASCII is a 7-bit code. Its arrangement is well thought-out, particularly in light of the technology available at the time it was designed. It was usually (and is now) transmitted as an 8-bit signal with the "most significant bit" (B7) reserved for parity—an error-detection scheme. For our purposes, we will set B7 to zero and concern ourselves only with B0 through B6.

					HEX	0	1	2	3	4	5	6	7
					B6-	0	0	0	0				
BIT	B3	B2	B1	B0	B5-	0	0	1	1	0	0	1	1
HEX					B4-	0	1	0	1	0	1	0	1
0	0	0	0	0		NUL	DLE	SP	0	@	P	`	p
1	0	0	0	1		SOH	DC1	!	1	A	Q	a	q
2	0	0	1	0		STX	DC2	"	2	B	R	b	r
3	0	0	1	1		ETX	DC3	#	3	C	S	c	s
4	0	1	0	0		EOT	DC4	$	4	D	T	d	t
5	0	1	0	1		ENQ	NAK	%	5	E	U	e	u
6	0	1	1	0		ACK	SYN	&	6	F	V	f	v
7	0	1	1	1		BEL	ETB	'	7	G	W	g	w
8	1	0	0	0		BS	CAN	(	8	H	X	h	x
9	1	0	0	1		HT	EM	)	9	I	Y	I	y
A	1	0	1	0		LF	SS	*	0	J	Z	j	z
B	1	0	1	1		VT	ESC	+	:	K	[	k	{
C	1	1	0	0		FF	FS	,	;	L	\	l	|
D	1	1	0	1		CR	GS	-	<	M	]	m	}
E	1	1	1	0		SO	RS	.	=	N	^	n	~
F	1	1	1	1		SI	US	/	>	O	_	o	DEL

Mnemonic	Meaning	Mnemonic	Meaning
NUL	Null	DLE	Data Link Escape
SOH	Start of Header	DC1	Device Control 1
STX	Start of Text	DC2	Device Control 2
ETX	End of Text	DC3	Device Control 3
EOT	End of Transmission	DC4	Device Control 4
ENQ	Enquiry	NAK	Negative Acknowledge
ACK	Acknowledge	SYN	Synchronous Idle
BEL	Bell	ETB	End of Transmitted Block
BS	Back Space	CAN	Cancel
HT	Horizontal Tabulation	EM	End of Medium
LF	Line Feed	SUB	Substitute
VT	Vertical Tabulation	ESC	Escape
FF	Form Feed	FS	File Separator
CR	Carriage Return	GS	Group Separator
SO	Shift Out	RS	Record Separator
SI	Shift In	US	Unit Separator
	DEL	Delete	

Table 1-2. ASCII (ITA#5) Coding

Note: To read the ASCII chart in table 1-2, follow these steps. There are seven bits in an ASCII character, B0 through B6, with bit 6 being the most significant. You will note that at the top of the chart in table 1-2 bits 4 through 6 are arrayed vertically. The lower four bits, bits 0 through 3, are arrayed horizontally. If you want to find a character, say, the uppercase A, go to the A in the chart. The column (vertical line) that the upper case lies in has B6 as 1, bit 5 as 0 and bit 4 as 0. Remembering that B7 is a parity bit and we have (for this part of the discussion) set it to zero, the upper four bits of uppercase A will be: 0 1 0 0 (Hex 4). To find the lower four bits, locate the row value (horizontal line) it lies in the row represented by 0 0 0 1 (Hex 1). Putting the two together results in the ASCII character code for uppercase A, which is:

0100 0001 (Hex 41)

Try finding the lowercase a: Vertically, it is 0110, and the horizontal row is the same 0001. Therefore, a lowercase a in ASCII character code is:

0110 0001 (Hex 61)

Note that the difference between upper- and lowercase characters is that B5 is a zero for uppercase characters and B5 is a 1 for lowercase characters.

Generally, if the most significant bit B6 is a 1, it is an alpha character, and B5 signifies whether it is upper or lower case, *that being the only difference* between the upper- and lowercase letter. If B6 is a zero, the character will be a numeric (0-9), punctuation, or a nonprinting control code. Further examination will show that if B6 is a zero and the next bit (B5) is a zero, the character is a control code. This means it is a nonprintable character. As ASCII was developed in the paper-tape days, one control character is stuck among all of the alpha characters. It is the DEL (delete) key, the all-ones combination. It was used to remove a paper-tape punch error.

Example 1-2:
Problem: Convert the following data into an ASCII-coded string. Note the use of HEX notation. (If you do not feel comfortable with HEX, you may use the binary representation for each character—but it will take up a lot more space.)

The quick brown fox. THE QUICK BROWN FOX.

Solution: Using table 1-2, we get:

T h e q u i c k b r o w n f o x.

54 68 65 20 71 75 69 63 6B 20 62 72 6F 77 6E 20 66 6F 78 2E

T H E Q U I C K B R O W N F O X.

54 48 45 20 51 55 49 43 4B 20 42 52 4F 57 4E 20 46 4F 58 2E

EBCDIC: Extended Binary Coded Decimal Interchange Code

This code was developed in the early 1960s by IBM and is proprietary to them. ASCII has only 7 bits, the eighth bit being reserved for parity. One problem in using only 7 bits plus 1 bit for parity arises when computers transmit program instruction coding. Computers normally operate using an 8-bit octet or a multiple of eight (i.e., a word that is 16 bits, or a double word of 32 bits). All 256 possible 8-bit combinations may not be used, but it is likely that the computer's instruction set would use the eighth bit. With the "7 information + 1 for parity" bit scheme the eighth bit isn't available. Most computers using ASCII transmit 8 bits and use a ninth bit for parity, if parity is used. To ensure that program code could be sent, IBM extended its 4 of 8 (BCD) coding, hence the EBCDIC name. This code is shown in table 1-3a and b. The blank spaces are used for special or graphic characters peculiar to the device using them. If required, this code can transmit "object" code—that is, the combinations of ones and zeros used to program a computer in 8-bit increments—with little

difficulty. EBCDIC did not require a parity bit for error correction, but instead used a different error-detection scheme.

One of the first interface problems of the PC age was how to perform PC-to-"mainframe" communications. By and large, most PCs and other devices transmitted in ASCII, while many of IBM's minicomputers and mainframe computers used a different coding—EBCDIC. This meant that they could not talk directly without some form of translator, generally a software program or firmware in a protocol converter.

	Most Significant Bits (B7, B6, B5, B4)							
Bits/Hex	**1000 8**	**1001 9**	**1010 A**	**1011 B**	**1100 C**	**1101 D**	**1110 E**	**1111 F**
LSBs B3,B2,B1,B0								
0000 0					{	}	\	0
0001 1	a	j	~		A	J		1
0010 2	b	k	s		B	K	S	2
0011 3	c	l	t		C	L	T	3
0100 4	d	m	u		D	M	U	4
0101 5	e	n	v		E	N	V	5
0110 6	f	o	w		F	O	W	6
0111 7	g	p	x		G	P	X	7
1000 8	h	q	y		H	Q	Y	8
1001 9	i	r	z		I	R	Z	9
1010 A								
1011 B								
1100 C								
1101 D								
1110 E								
1111 F								

Table 1-3a. EBCDIC (Upper 8 MSB)

	Most Significant Bits (B7, B6, B5, B4)							
Bits/ Hex	**0000 0**	**0001 1**	**0010 2**	**0011 3**	**0100 4**	**0101 5**	**0110 6**	**0111 7**
LSBs B3,B2, B1,B0								
0000 0	NUL	DLE	DS		SP	&	`	
0001 1	SOH	DC1	SOS					
0010 2	STX	DC2	FS	SYN				
0011 3	ETX	DC3						
0100 4	PT	RES	SYP	PN				
0101 5	HT	NL	LF	RS				
0110 6	LC	BS	ETB	UC				
0111 7	DEL	IL	ESC	EOT				
1000 8		CAN						
1001 9	RLF	EM						\
1010 A	SMM	CC	SM		@	!	|	:
1011 B	VT				.	$	,	#
1100 C	FF	IFS		DC4	<	*	%	&
1101 D	CR	IGS	ENQ	NAK	(	)	_	‘
1110 E	SO	IRS	ACK		+	;	>	=
1111 F	SI	IUS	BEL	SUB		-	?	“

Table 1-3b. EBCDIC (Lower 8 MSB)

Example 1-3:

Problem: Convert the following data into EBCDIC. Use table 1-3a and b. Use HEX notation for brevity.

The quick brown fox. THE QUICK BROWN FOX.

Solution: Using table 1-3, we get:

T h e q u i c k b r o w n f o x.

E3 88 85 40 98 A4 89 83 92 40 82 99 96 A6 95 40 86 96 A7 4B

T H E Q U I C K B R O W N F O X.

E3 C8 C5 40 D8 E4 C9 C3 D2 40 C2 D9 D6 E6 D5 40 C6 D6 E7 4B

Unicode

Herewith a brief but succinct history of data communications. Character coding electrically became established with the Morse code, in which letter frequencies and a form of compression were used for efficiency. Fixed-length character representations, of which ASCII and EBCDIC are examples, came about in the early 1960s and are still in wide use today. As we move forward, however, a different character coding is appearing: Unicode. Instead of representing a character with 7 or 8 bits, it uses 16 bits. For users of ASCII this is not a problem (as long as the extra eight more significant 8 bits are accounted for) because the Unicode for ASCII is HEX 00 + ASCII. In other words, the upper 8 bits are set to zero, and the ASCII character set follows. The following is excerpted from Microsoft Corporation's Help file for Visual Studio 6 (MSDN Library April 2000):

"Code elements are grouped logically throughout the range of code values, which is called the codespace. The coding begins at U+0000 with standard ASCII characters, and then continues with Greek, Cyrillic, Hebrew, Arabic, Indic, and other scripts. Then symbols and punctuation are inserted, followed by Hiragana, Katakana, and Bopomofo. The complete set of modern Hangul appears next, followed by the unified ideographs. The end of the codespace contains code values that are reserved for further expansion, private use, and a range of compatibility characters."

This language is disseminated under the auspices of the Unicode Consortium. It does not dictate how the character should appear, but merely that it should be processed as this character. The printing is up to the software and hardware. Most modern office suites

(Microsoft Office 2000-2007 and Open Office 2.0 in particular) are set up to handle Unicode. One must remember that this is an international world, made much smaller by data communications, and having a common language standard (for processing purposes) is a good starting point for communicating with that world.

Data Organization: Error Coding

How we organize our data has a lot to do with how we recognize and correct transmission errors. These are alterations to the intended data that occur, due to electrical noise for example, as our signals traverse the transmission media. We cannot correct errors before we place them on the media; we assume the data to be transmitted is correct. If that is not the case, we need a whole different set of error-detection techniques that are far beyond the scope of this text. And we can only detect errors after they have been taken off of the media (received). The IBM 4 of 8 code was an example of early error detection. If any character had other than four 1s in 8 bits, it was in error. The entire block (84 characters) would then be retransmitted. Parity is an extension of this concept.

Parity

Parity is a means of error detection. While using 7 bits of an 8-bit structure, as ASCII does, will leave 1 bit extra, in most cases B7 is set to zero, and the parity bit is added to the 8-bit character, making it 9 bits. Parity (in the telecommunications sense of the word) means to count the number of ones in a character (zeros could be counted, but traditionally only the ones were). The agency determining the system's operating specifications will have decided on which parity to use, that is, "odd" or "even." If "odd" parity is used (see table 1-4), then the parity bit will be whatever value is required to ensure an odd number of ones in the character (including, of course, the parity bit).

ODD PARITY SELECTED				
	CHAR	**CHAR**	**CHAR**	**CHAR**
	A	B	C	D
Parity	1	0	0	1
B7	0	0	0	0
B6	0	1	0	1
B5	1	0	0	0
B4	1	1	1	1
B3	0	0	1	0
B2	1	0	1	0
B1	1	0	0	0
B0	0	1	0	0

Table 1-4. Character Parity

Table 1-4 illustrates how characters' parity is determined. Note that the characters are

arranged vertically. This form of parity is called "vertical parity," for reasons that will become evident momentarily.

Example 1-4:
Problem: Using odd parity, add the correct parity bit to the following ASCII string.

What is your name?

Solution: Using the ASCII in table 1-2, determine the bit combinations of each character. Add a 1 as the parity bit to ensure that the total of 8 bits (vertically) has an odd number of ones.

BIT	W	h	a	t		i	s		y	o	u	r		n	a	m	e	
Parity	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**1**	**1**
B7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B6	1	1	1	1	0	1	1	0	1	1	1	1	0	1	1	1	1	0
B5	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
B4	1	0	0	1	0	0	1	0	1	0	1	1	0	0	0	0	0	1
B3	0	1	0	0	0	1	0	0	1	1	0	0	0	1	0	1	0	1
B2	1	0	0	1	0	0	0	0	0	1	1	0	0	1	0	1	1	1
B1	1	0	0	0	0	0	1	0	0	1	0	1	0	1	0	0	0	1
B0	1	0	1	0	0	1	1	0	1	1	1	0	0	0	1	1	1	1

Block Parity

Table 1-5 illustrates why the characters in the last section were arranged vertically. Because punched cards (at least in IBM's version) contained 80 columns, and historically most data was on these cards, transmissions were in an 80-character "block." Two framing characters were added at the start of the block, and one framing character was added at the end of the data block. An eighty-fourth character, called the block parity character, was added to the end of the transmitted block. It is computed from the other 83 characters.

	SOH or STX	SOH or STX	1st Char.	Information of 78 characters	Last Char.	ETB or ETX	Block Parity
Parity	0	0	1		1	0	1
B6	0	0	1		1	0	1
B5	0	0	0	**78**	0	0	1
B4	0	1	0	**CHAR**	1	0	1
B3	0	0	0	⟷	1	0	0
B2	0	0	0		0	1	0
B1	1	0	0		1	0	1
B0	0	0	1		0	0	0

Table 1-5. Block Parity

Parity was first determined vertically for the block's 83 characters. Parity was then determined horizontally along the rows of each column, *including the parity row*, for all 80 counted columns.

The results of the horizontal parity were placed in the corresponding row bit of the block parity character. For the block parity character, the vertical and horizontal results must match in order to correctly determine the parity bit of the block parity character. If not, then there was an error. The same parity scheme used for vertical parity must also be used horizontally. This scheme is known as a vertical and horizontal parity checking or "block" parity.

In block parity, fewer than 1 in 10,000 errors can go undetected. However, this scheme extracts a heavy penalty in overhead. One parity bit for each character adds up to 84 bits, or more than ten eight-bit characters.

Error Correction

Besides the number of bits devoted to detection, once an error was detected what means could be used to correct it? Usually, the answer was *Automatic Request for Repeat* (or Repetition, or Retransmission) known as ARQ. This method had significant ramifications in many applications since the transmission device had to store at least the last transmitted block. There also had to be some scheme to notify the transmitter when the receiver had either successfully received the transmission or to retransmit the block in error. In most cases, half-duplex is too inefficient in terms of transmission time versus line-setup time (i.e., the time it takes for the transmitting device and receiving device to establish connection and synchronization).

A number of schemes have been devised to increase the "throughput" (defined here as the actual number of bits correctly received at the end device). In modern usage, many schemes use a block that varies in length, depending on the number of errors detected. In transmissions that have very few errors (such as fiber optics or local area network, or LAN, types), the block length could be made longer. In media with large errors the block could be made shorter. Blocks are generally of a fixed length, or a multiple thereof. Packets, on the other hand, will have a fixed minimum and maximum length, but can vary in length between these two limits. We normally speak of packet transmission in modern data communications, although the difference between a block and a packet is more one of semantics than practice. Note that the longer the block/packet the greater the chance of an undetected error. Parity and block check schemes were far less effective with multi-bit errors versus single-bit errors.

Cyclic Redundancy Checks

Most modern devices use a far more efficient scheme of error detection than parity checking. Though ASCII was designed for a vertical and horizontal (at times called "longitudinal") block-checking code, it can also be used with a *cyclic redundancy check* (CRC) scheme. If parity checking is not used, then the eighth unused bit in ASCII may be used for graphics. Alternatively, octets of computer program code may be sent.

In a CRCC, a cyclic code divides the text bits with a binary polynomial. That is, it samples every bit in a serial stream and combines data from selected (and fixed) bit positions. Figure 1-6 illustrates the generation of a simple CRC. They are sometimes called cyclic redundancy check characters (CRCC) or (incorrectly) checksum. A number of different CRCCs are used today, but their primary difference is the pick-off point (or power of X in the representative equation such as the CRCC-CCITT that uses $G(x) = X^{16} + X^{12} + X^5 + 1$). The advantages of one pick-off point or another depends on the application. A communications channel may have different error behavior than a magnetic media. In any event, the transmission protocols used in industrial data communications will probably use the CRCC-32 (a local area network CRC of 32 bits).

Within a given size block (packet or frame), a certain size character or characters will be generated. A common (CRC-CCITT) type uses two 8-bit octets, forming a 16-bit check character. The characters are generated during transmission, and the block is stored. The receiving end receives the text and computes its own CRCCs. These must match the transmitted characters or there is a detected error. Note that even on a block of 80 characters this scheme only uses 16 bits, compared to the 88 used with the vertical and horizontal parity check. This is the method used when writing to a disk drive or to most any magnetic media, and it is the method most often used (whether the CRC is 16 bit or 32 bit) in modern data communications.

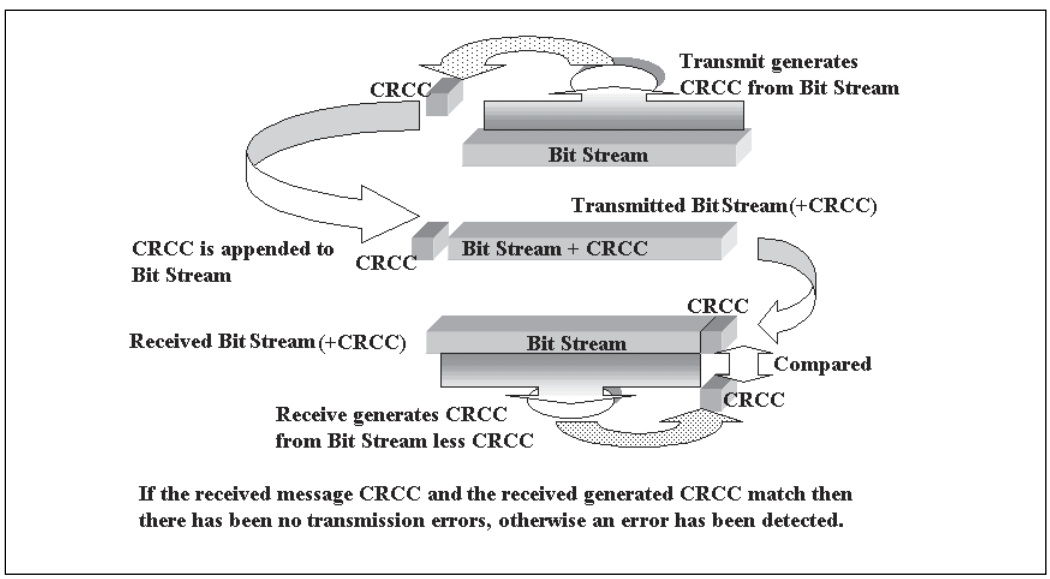


Figure 1-6. CRCC Concept

Checksum

Any of a number of error-detection codes may be called a *checksum*. Many times the CRC-16 or CRC-CCITT is called a checksum, but they are actually cyclic codes, whereas checksum was originally intended as a linear code. For instance, all the 1 states in a block

may be totaled, and this number (usually through modulo addition) tacked on as a block check character or characters. The efficiency of these checksums in detecting errors is not as high as a cyclic code, but the circuitry to produce the checksum is less complex.

Other codes exist specifically for detecting and correcting errors. Some are cyclic, some are linear, some are quasi-cyclic, and other types of polynomial codes are used for error detection. Some of these codes are sufficiently long (in pattern length) to be used for error detection and correction.

Forward Error Correction

Though the ARQ method of error detection discussed earlier detects an error (by whatever means) and then retransmits the portion of the message that was in error, what happens when there is simplex (unidirectional) transmission? The ARQ method offers no way to receive an acknowledgment to retransmit. In this case or in some cases where the medium is too noisy to allow any significant throughput, *"forward" error correction* (FEC) may be employed. This requires that the communications network's error disturbances be analyzed and an error-correcting algorithm be designed to fit the error patterns. This can be done for most real-world circuits without extraordinary difficulty. The main problem is that the error-locating overhead can be as much as one parity (or error-detecting symbol) bit for each data bit. This would immediately cut throughput in half. In some cases, however, the data must be exact (without regard to capital expense or throughput), since this transmission may be the only means or the only time, given the circuit, that any throughput will occur. (Think real-time data, satellite tracking data, nuclear plant critical monitoring, and so on.)

A thorough discussion of error-correcting codes and associated theory is beyond the scope of this text. Many good references, at many different knowledge levels, are available on this subject, however. Knowledge of error correction is integral to data communications—but only in regard to its implementation, not necessarily with respect to the theory behind it.

Data Organization: Protocol Concepts

A protocol is an accepted procedure. We use protocols in personal interactions all the time. In the United States, when one is introduced to someone, the protocol is to shake hands, unless there is a ceremony (requiring another form of protocol) or you are being introduced to a large number of people. In a class, or indeed throughout a conversation, it is protocol that tells you that you should wait until another is finished speaking before speaking yourself. There is a reason for this protocol (other than to keep you from speaking while I am): humans are mostly half-duplex; typically, they cannot very well talk and hear at the same time.

All communications efforts involve protocols of one kind or another. Such protocols are a higher level of data organization than placing ones and zeros into characters. Communications protocols are either character based or bit oriented. This means that the information we are looking for will take the form either of characters telling us information or bit patterns (other than characters) telling us information.

Asynchronous versus Synchronous

Asynchronous generally means that something may occur at any time and is not tied to a clock. The old "start-stop" teletypewriter signal (see appendix 5) with its one start bit and one (1.45 or 2) stop bit is a good example of asynchronicity. The teletypewriter signal started out using motor speed as the main element synchronizing the start-stop bits with each character. In today's vernacular, any start-stop signal is assumed to be asynchronous. In actuality, a start-stop signal may be transmitted synchronously or asynchronously.

Synchronous generally means tied to a common clock. Typically, the clock signal is transmitted along with the data, usually in the transitions of data from one state to another. Since synchronous transmission uses bit timing, each bit of data must be accounted for. Almost all modern data communications are synchronous transmissions.

The terms *baud rate* and *bits per second* are often used interchangeably; this is incorrect. Baud is a line modulation rate, that is, the number of signal changes per second that are being placed onto the media. (The maximum value for this is called the 'bandwidth' of the channel.) Bits per second (bps), on the other hand, is the data transmission rate of the device that is transmitting or that the device is capable of receiving. Depending on the type of signaling and encoding used, the baud rate and bit rate can be quite different.

Example 1-5:

Problem: How to determine the line modulation rate if a device transmits a rectangular pulse that is 1/1200 of a second but occurs only once per hour (1 bit per hour).

Solution: Its bit rate is one pulse per hour. However, the media must be capable of passing a rectangular pulse of 1/1200-second duration. To determine the baud rate, divide the *shortest information element* (for noncoded signals) *into 1*. The interval for 1/1200 of a second is 0.0008333; dividing this into 1 gives, of course, 1200. The baud rate required is 1200.

Baud per second is a term that is used to describe a change in a transmission signaling state, not a line speed. Ads for a 33.6Kbaud modem would actually mean that the modem requires a line with a bandwidth great enough for 1/33600-second rectangular pulses. What they mean in reality; these are 33,600-bits-per-second (33.6Kbps) modems (the data bit rate). These modems use a 1200 baud line (the typical telephone line) and achieve their higher data rate by sending 56 bits of data each baud. That is they make one signal change for the state of a 56 bit piece of data. This is accomplished through high-level coding techniques called trellis encoding. These modems require a 600-baud line each direction (a total of 1200) rather than one that supports 33,600 signal changes a second in each direction

which is what a standard dialup telephone line can support. Some refer to the baud rate as symbols per second.

As electronic message handling became the norm, it became imperative to build as much of the link control into the terminals as possible. To this end "communications protocols" arose. Most protocols were developed by individual vendors for their own systems. Because these protocols are generally incompatible with the protocols and equipment of other vendors, the customer is generally locked into one manufacturer, promoting incompatibility.

Character-Based Protocol

One example of a character-based protocol is Binary Synchronous Communication (Bi-Sync) (another is the teletypewriter system, see appendix B). One of the first and most widely used of the proprietary protocols is the IBM Bi-Sync protocol, which was developed to link the IBM 3270 line of terminals to computers in a synchronous manner. (The protocol may also be used in a system with asynchronous signaling, that is, with start-stop characters, provided the text mode is used). The IBM Bi-Sync code is character-oriented: that is, control depends on certain character combinations rather than on bit patterns. It also requires that the transmitting terminal be able to store at least one block of data while transmitting another.

In transmitting Bi-Sync, the hardware is responsible for avoiding long strings of ones or zeros. In a synchronous system, if a significant length of time is used to transmit only one state or the other, the receiver loses its bit synchrony. As a result, communication would be disrupted or, at least, in error. Most modern hardware generally uses a scrambler to ensure data transitions. The scrambler contribution is removed at the receiver since the scrambler pattern is performed on a scheduled basis. Table 1-6 lists some of the Bi-Sync control characters.

SYN	Synchronous Idle	Used to synchronize receivers
SOH	Start of Header	Indicates routing information
STX	Start of Text	Indicates message text starts
ETX	End of Text	Indicates end of transmitted text
ITB	End of Intermediate Block	More blocks coming
ETB	End of Transmission Block	Do error count, block over
ACK	Acknowledgment	Received block OK
ACK1	(same as ACK)	Received odd block OK
ACK2	Acknowledgment 2	Received even block OK
NAK	No Acknowledgment	Bad block, retransmit
ENQ	Enquiry	Go ahead and send
DLE	Data Link Escape	Pay no attention to control characters until a DLE pair appears again
EOT	End of Transmitted Text	This transmission has ended

Table 1-6. Bi-Sync Control Characters

Example 1-6:

Problem: Show the Bi-Sync sequences required to transmit the following message: This is a short block. Just indicate where their presence will be in the message stream. No pad characters are included to produce a minimum block size). ASCII in HEX notation is used.

Solution:

```
16  16    01   41  02   54 68 69 73  20  69 73 20 61  20  73 68 6F 72 74
SYN SYN  SOH  A  STX   T  h  i  s        i  s       a       s  h  o  r  t
20 62 6C 6F 63 6B 2D   04   17  BCC1 BCC2  16  16
b  l  o  c  k  .       EOT  ETB CRC1 CRC2 SYN SYN
```

The receive portion of the circuit that just transmitted will now wait for an ACK (06) before proceeding, or if a NAK (15) occurs it will retransmit this block of text.

Bi-Sync uses full-duplex transmission. Though it could use half-duplex, it was not primarily intended for that type of operation. Prior to Bi-Sync, a block would be transmitted, the line would be turned around, and the transmitter would wait for the receiver to send either an ACK or a NAK. If a NAK was received, the transmitter would retransmit the block. Full-duplex operation would not have been faster (except for eliminating the turnaround time) because the transmitter could do nothing else until it received a response to its transmitted block. To speed things up a bit, the Bi-Sync had the transmitter store two blocks so it could wait for an ACK while transmitting the second block. ACK1 and ACK2 signals were used to differentiate between ACKs. The primary benefit in this case is that transmission could take place without line turnaround (duplex) or without halting transmission until an ACK was received.

Because the control characters are intertwined with the text, they must be sent in pairs to ensure they are identified. How would you transmit a machine or computer program in object (machine-executable) form if that code were composed of 8-bit octets, some of which may very well be the same as the control codes? To make this possible, Bi-Sync allows a transparent mode in which control characters are ignored until the receiver detects several DLE (Data Link Escape) characters.

Bi-Sync is dependent upon character-oriented codes. In modern communications there is a need to make the protocol independent of the transmitted message type. That is, it should make no difference to the protocol what bit patterns the message consists of or even in what language it is composed, as long as it is in 8-bit octets. LAP-B (described later in this chapter) and other bit-oriented protocols accomplish that functionality rather gracefully.

Framing

Bit-oriented protocols use a concept called "framing." In framing, there will be a binary pattern (which the protocol ensures cannot occur in the bit stream) that will be the start delimiter (starting point). There will also be binary patterns that indicate the addressing and what type of frame this is (e.g., it contains information, or it is of a supervisory nature), and some method of sequence numbering followed by the user data. A Frame Check Sequence, normally a CRCC, will follow the user data which is typically a variable number of octets. The user data is surrounded by the protocol; the protocol "frames" the user data (see figure 1-7). There may also be a stop delimiter, or the frame may use the CRCC as the delimiter.

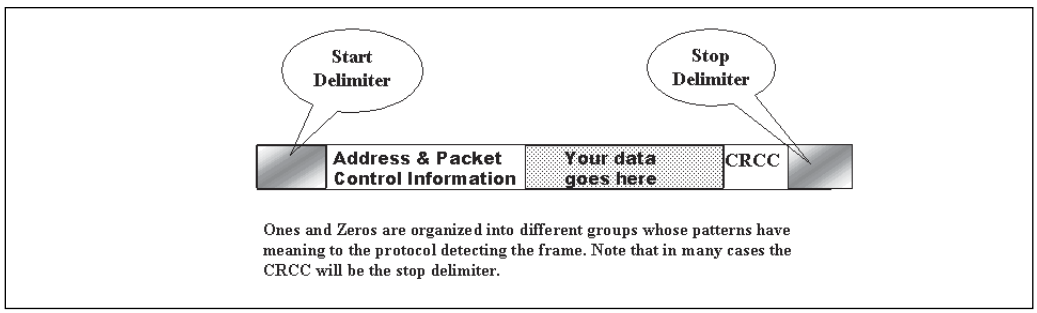


Figure 1-7. Frame Concept

Link Access Protocol (LAP-B)

Link Access Protocol-Balanced (LAP-B) is a bit-oriented protocol. It is very similar in both structure and format to other bit-oriented protocols: High Level Data Link Control (ISO HDLC), Advanced Data Communications Control Procedure (ANSI ADCCP), and IBM's Synchronous Data Link Control (SDLC). IBM uses SDLC, which is a subset of HDLC, in its Synchronous Network Architecture (SNA). ADCCP and HDLC are quite similar. For that reason, only LAP-B will be discussed here.

Figure 1-8 illustrates a LAP-B frame. Notice that it is bounded by "flag" characters. In other words, the flags "frame" the data. The flag is an 8-bit octet, starting with a zero followed by six ones and ending with a zero. It is inserted at the beginning and end of each transmitted frame. The protocol only allows the frame to have this pattern at its start and end. It does so by using a technique called "zero insertion" or "bit stuffing."

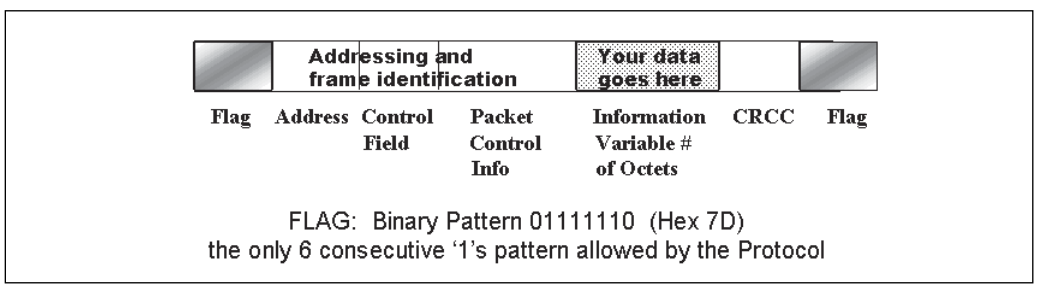


Figure 1-8. LAP-B Frame

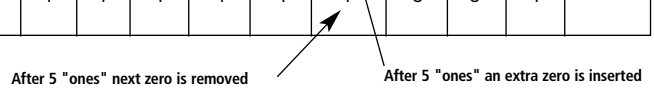
Zero Insertion

In normal transmission, any time the protocol detects five 1 bits in the data stream, the rule is to insert a zero. The receiver protocol, upon detection of five consecutive zeros, knows to remove the zero that follows. It is that simple. Example 1-7 illustrates zero insertion and removal.

Example 1-7:

Problem: Show the zero insertion and removal process for the following bit stream:

Bit Stream	0	1	1	0	1	1	1	1	1	1	0	0	1	
Zero Inserted	0	1	1	0	1	1	1	1	1	**0**	1	0	0	1
Data at receiver	0	1	1	0	1	1	1	1	1	0	1	0	0	1
After removal	0	1	1	0	1	1	1	1	1	1	0	0	1	



Example 1.8. Zero Insertion

In LAP-B, the protocol will allow reception of up to 7 or 128 frames, depending on the application, before it must have the first frame acknowledged. Media errors are detected through the frame check sequence (FCS) and identified by frame number.

An error will destroy that frame number. The transmitter receives notice because the destination receiver ignores an errored frame and notifies the transmitter upon receiving the next good frame that it is not the one it anticipated. The receiver will request either retransmission of the damaged frame and perhaps all subsequent frames, depending on what type of system is in operation. Note that the information carried in the frame has no bearing on the protocol.

We will discuss bit-oriented protocols in more detail in both chapters 2 and 7.

Protocol Summary

We have looked at two protocols, one character-oriented and one bit-oriented. We have left out much of the inner workings and subsequent details from the discussion in order to illustrate only the salient points of some protocol concepts. Actually, as you proceed through the next chapter, you will find that these were Data Link protocols, and so these bit-oriented protocols will again be discussed in chapter 2. For this chapter, it is only important that you see how ones and zeros are organized in order to frame the actual characters you are transmitting.

First, the bit protocols do not depend on the data stream contents; they operate independently. This means that the protocol recognizes certain bit patterns that the protocol does

not allow to occur in the data stream. Second, LAP-B operates to frame and error-check packets. LAP-B may be used in any system, but it finds use primarily in point-to-point and multi-drop systems.

Third, framing the data usually means having some form of start delimiter (flag), some sort of addressing and control process, the actual user data, an error check on the frame, and a stop delimiter (if the error check is not the stop delimiter itself). The user data is then said to be "*encapsulated*." Figure 1-7 illustrated the general frame. Note its similarity to the LAP-B frame in figure 1-8.

Summary

In this chapter we have reviewed the organization of data: data as organized into characters (for transmission, storage, and presentation to humans) and as organized to detect errors in transmission. The first data transmission code (IBM's 4 of 8) devoted more overhead to the error-detection scheme than to the data transmitted. Use of cyclic redundancy codes has minimized the necessary overhead while enhancing the accuracy of error detection.

More than anything, this chapter has served to introduce you to the foundational concepts of data communications and how it is organized in modes of transmission, in character codes, and in protocols. In the following chapters, we will organize data further into information (protocols and such) and simply implement what we have discussed in this chapter. The key point to acquire from this chapter is that data, ones and zeros, means nothing unless it is organized into some accepted structure that then enables it to become useful information.

Bibliography

Please note that when Internet references are given the address was valid at the time of the chapter creation. Web sites come and go so occasionally one of the references will no longer work. It is best then to use a search engine to locate the topic. The web addresses are given to provide credit for the information referenced.

Keogh, Jim. *Essential Guide to Networking*. Upper Saddle River, NJ: Prentice Hall, 2001.

Microsoft Corp. *Unicode*. MSDN Library, April 2000.

Peterson, W. Wesley, and E. J. Weldon, eds. *Error Correcting Codes*, 2d ed. Boston: MIT

Press, 1988.

Sveum, Myron Even. *Data Communications: An Overview.* Upper Saddle River, NJ: Prentice Hall, 2001.

Thompson, Lawrence. *Electronic Controllers.* Research Triangle Park, NC: ISA, 1989.

Thompson, Lawrence. *Industrial Data Communications*. 3d ed. Research Triangle Park, NC: ISA, 2001.

2
Communications Models

Modeling

Before explaining the models used in data communications, it might be useful to explain what models are and what purpose they serve. A model is a simulation of a real object. It may be a mathematical description, it may be an analogue of the original (a model of the object's physical characteristics), a set of logical constructions, or even a set of functions. A model can contain all of these properties. We use models to simplify explanation, when the real object is too elusive to use in human terms and to represent objects that we cannot physically capture.

In communications we use models of different types to explain functional or circuit operation and to design communications. In this chapter we look at six models. We will first consider the International Organization for Standardization (ISO) Open Systems Interconnection (OSI) model, which is a model of the functionality required to communicate from end user to end user. The second model we look at is the Institute of Electrical and Electronic Engineers' (IEEE) 802 LAN model. We then close the chapter by discussing a set of four modern applications models. To understand data communications, it is important that you grasp what these models represent, as almost all discussions [both in this book and generally] of protocols and standards are based on these models.

ISO OSI Model

How are the groups of ones and zeros, the data's organization, transmitted from one end user to another? It is not enough that data be organized; there must be a system for moving this data from one location to another. This system is comprised of rules, rules about who has access to the media and when (media access), who does packet (frame) error detection, who counts packets, who performs routing, who is responsible for end-user-to-end-user communication, who keeps track of the variety of traffic, who ensures that the traffic is compatible with the host or remote stations, and who interfaces with the user program. All of these functions have to be performed when communicating from one end user to another. The OSI model describes these functions, and generally specifies the order in which they take place in transmission. The OSI model is complex, and understanding its functions is crucial to understanding data communications.

Yet the actual information needed to gain this understanding is actually quite simple. Though most readers have probably been exposed to the OSI model, it is usually explained only in brief statements or in the implementing standards themselves. One reason for this is

that this is a model of function, not a hardware or a software specification. If a system is OSI compliant, then particular OSI standards are implemented in that layer. Other standards may perform a particular layer's functions, but that does not make the implementation OSI compliant. The best way to begin an explanation of the OSI model is with an analogy.

Mail Analogy

Figure 2-1 is a simple analogy of communications functions. In the "good" old days (before businesses computerized) if you wished to send a purchase order (PO), how did you do it? First, your research had given you the information on the technical specs of the product you wanted and perhaps its manufacturer too (if your company did not have equipment preferences).

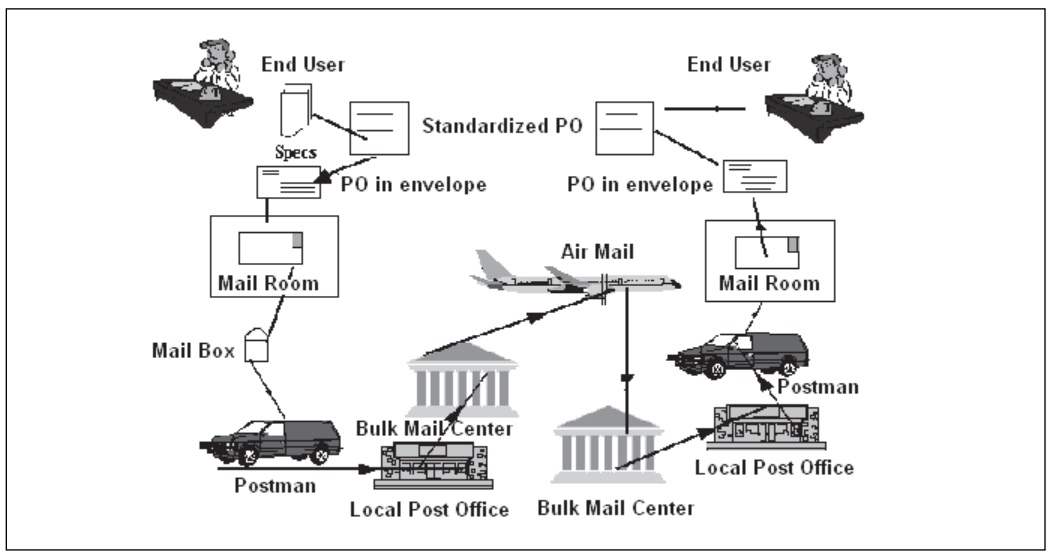


Figure 2-1. Mail Analogy

Next, you filled out a standard requisition form. This means you put the data into a standard format acceptable to both your company and the vendor. This form was usually typed by a secretary. It was then determined how many copies and to whom it should be addressed within the vendor company, and all this information was attached to the PO.

Then, a mailing label was produced (by a secretary or perhaps in the mail room), and the PO was placed in an envelope for shipping. Note that at this point the information on the PO is irrelevant; only the address on the label is important. The mailroom placed postage (and perhaps the label) on the envelope and dropped it in the box outside for pickup. The mailroom attendant didn't care what was in the envelope, only where it was going and how much it weighed. The mail truck picked up the mail and took it to the local post office (the last two digits of a five- digit zip code in the United States). The employees at the post office were only concerned with the first three digits of the five-digit zip code, for that is the address of the receiving bulk mail center, and if the item was not meant for delivery within the local zone it went to the regional bulk mail center, typically by truck. The

regional bulk mail center determined where the item was to go next, based only on the first three digits of the zip code. It might go by plane, train, or bus to the receiving regional bulk mail center in that zone.

At the receiving regional bulk mail center your PO went to the local post office, from there to the vendor's mail stop, to the vendor's mail room, to the mail entry (or similar position) clerk, and then finally to someone who read the PO to determine what you required.

Since the moment you placed your vital information in the envelope, no one cared what that information was until it was received at the vendor by the person who could act on the PO, the end user. That is, information was added (to the envelope) and utilized without any regard to what was in the envelope as long as the envelope was of standard dimensions and weight. Conversely, when you filled out the purchase order, you did not consider how many places the envelope that contained it would be picked up at and delivered to on its way to the end user.

Without carrying this analogy too far, it does bear a close similarity to the OSI model for end-user-to-end-user communications. One very critical question should be addressed here—how did the user know that your letter was received? The end user who actually looked at the contents acted upon that information. This response is analogous to the "connectionless-oriented" transmission or Type 1 (OSI) method. You send your data (encapsulated or "framed" by the necessary overhead), and regardless of the number of times it is handled (these instances are sometimes referred to as "hops" when referring to Internet communications) you receive no confirmation at the receiving end. Perhaps a better way to envision this is to send multiple letters at one time. They arrive at different times and probably in random order. A user confirmation when the message has been totally received may be by way of acknowledgement (Type 3) or handled at a different layer of functionality. Type I transmission (no acknowledgement) is referred to as a datagram.

If you must ensure that your mail has progressed each step of the way, then you send a registered letter. Every time it is handled, it must be signed for. In many ways this is analogous to "connection-oriented" transmission or Type 2 (OSI). One very similar effect to registered mail is that Type 2 almost always takes much longer to arrive than Type 1 just as registered mail takes longer than the regular mail. Or, just take the example of a dial-up telephone call. You establish the connection before any transmission takes place and the circuit remains connected (hopefully) for the length of the transmission. (I am indebted to George Stiefelmeyer for the postal example. Although I have taken a few liberties with it, I have always found it to be quite effective in explaining the OSI model.)

OSI Model

The Open Systems Interconnection reference model is an attempt to standardize the functionality of end-to-end computer communications. Some communications systems built today are OSI compliant (meet all the open standard specifications). However, the large

majority of systems are not OSI compliant, though they do implement the functionality in one or more of the OSI layers. As an example, the Internet Protocol (IP of TCP/IP fame) is not OSI compliant, yet it is described as a Layer 3 protocol.

End-to-end computer communications encompass many disparate systems. What is defined in the OSI model is not hardware but a set of functional layers. The OSI model has seven layers (see figure 2-2). It makes no attempt to specify any modem, media, medium access, or any of the physical standards such as connectors, coax, and so forth. Rather, it allows existing standards that meet the OSI functional requirements to fit into place.

The line of demarcation between data communications and data processing exists at the border between the Transport and Session layers. However, many systems with both these functional areas integrate them, so defining the layer boundary is quite difficult. Note that the User Program shown in figure 2-2 is not a layer in the OSI Model. It is included merely to illustrate the connectivity paths.

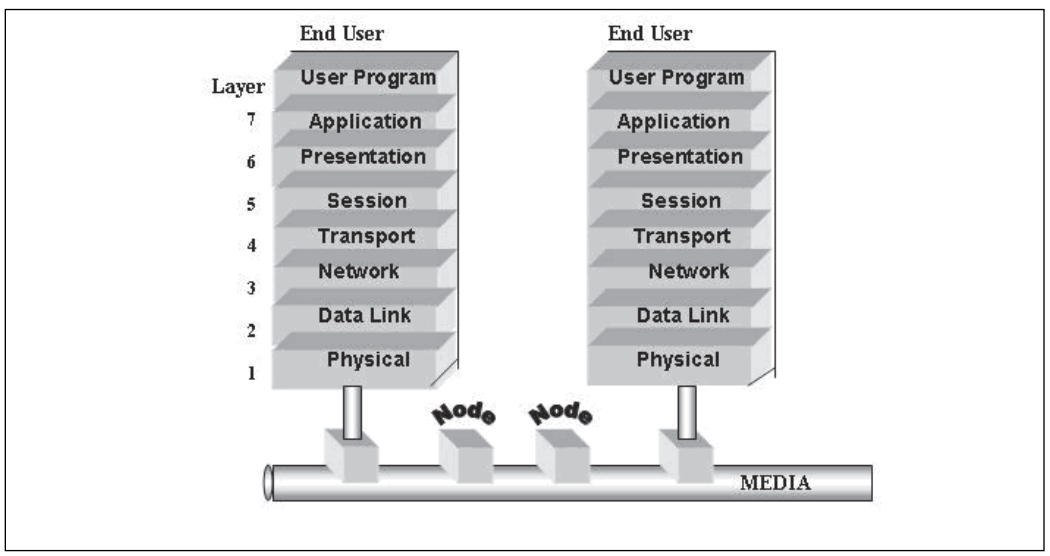


Figure 2-2. ISO-OSI Model of Interconnection

Physical

The Physical layer *provides the actual means of connection to the media, be it copper, fiber, or wireless*. Some examples of the Physical layer include EIA/TIA 232, EIA/TIA 485, or the LAN Network Interface Card (NIC), where it provides line termination and/or impedance matching as well as synchronization of data (all of which are discussed in later chapters). The interfaces between layers, such as between Layer 1 and 2, are known as Service Access Points (SAP). For every connection (or task requiring communications) there will be a set of SAPs: the Source Service Access Point (SSAP) and the Destination Access Service Point (DSAP). If you have duplex operation there will be two sets of SAPs.

SAPs are nothing more than addresses in memory assigned by whatever program is controlling the communications. They are well defined in the IEEE 802 series, which establishes local area network standards. If yours is a multitasking machine, more than one communications task may very well be in process. Hence, a number of different SAPs may be active at one time.

Data Link Layer

The *Data Link layer frames (encapsulates) data* and provides "error-free" transmission to the Network layer. This layer always ensures that Frame Check Sequences (FCS) check for and request retransmission of erroneous frames. For a Type 2 transmission (connection-oriented), the Data Link layer also checks for missing message sequences and lost or duplicated frames (packets). Into this layer would fit such protocols as Bi-Sync, SDLC, ADCCP, HDLC.

Network

The *Network layer performs end-user-to-end-user routing*. Specifications that fit this layer include CCITT X.25 or parts of various LAN operating systems as well as any functions associated with data communications network routing and control. Layer 3 is almost totally focused on routing functions. The Internet Protocol (the IP of TCP/IP) and Internet Packet Exchange (IPX of Novell fame), along with the OSI Internet protocol, are all concerned with routing and the path that packets will take from end user to end user. All fit into the Layer 3 functionality model.

Transport

The *Transport layer is responsible for reliable end-user-to-end-user communications*. It translates the lower-layer information into a data processing format, and places the 1s and 0s into packets in the downward (toward Layer 3) direction. In a number of connection-oriented systems, the Transport layer is null (not used). Why? Because all of the error checking, both transmission and frame assembly and disassembly, packet counting, and sequencing (frame and packet are sometimes used interchangeably) are performed in Layer 2. "Connection-oriented" implies that a connection has been established prior to data transmission. Such is the case with many current industrial protocols using only Layers 1, 2, and 7. Transmission Control Protocol (TCP of TCP/IP fame) is called a Layer 4 protocol. As a protocol it is actually connection-oriented (UDP is connectionless) even if the underlying system is connectionless (such as the Internet). It is not OSI compliant, yet it has Layer 4 functionality and is the de facto standard for communications today.

Session

The Session layer is concerned with the jobs at hand and the scheduling of jobs from the Applications layer (through the Presentation layer). It will allocate system resources as required and communicate commands to the Transport layer. The Session layer establishes virtual connections and closes them when the communication tasks are complete. You do much the same thing when you make installment-type payments through the mail. You mail the first payment (open a connection) and transfer payments until the note is paid off.

If you are like many people, you make a number of these communications monthly. You don't know the actual route to the end user (note holder) nor do you know it has arrived except when it doesn't. You cease communications and close your session when you receive the note papers. You are essentially doing what the Session layer does, only the Session layer does it in a much shorter period of time.

Presentation

The Presentation layer ensures that the communications data is typed correctly for the Applications layer. Not all systems employ one type of computer architecture. Even among compatible machines, some are (were) 8-bit, some 16-bit, some 32-bit, and some 64-bit. Each machine has a different set of rules. The Presentation layer ensures that communications is performed in a common language; the individualization of the data is also performed at the Presentation layer. This layer ensures that all the data typing and formatting will interface with the Application and Session layers. The Presentation layer is where encryption and decryption of data is accomplished. One of the details that has yet to be worked out at this level is ensuring that the data format is the same, that is, big-endian versus little-endian. If the most significant bit (from the perspective of a binary data value) is transmitted first, followed by all other bits to the least significant bit, it is called big-endian. Little-endian (used by Intel™ and others) transmits the least significant bit first, followed by all other bits to the most significant bit. It should be obvious that for any meaningful communications to occur, the bit formats should be the same. The mismatch is easy to correct, but it must be identified first.

Application

The Applications layer is where the system meets the user. Applications layer functions are extremely high level compared to the bit functions at the Physical layer. The application in question might take the form of a process controller, a database manager, or any of the myriad of user applications. The Applications layer is much like an operating system using a graphical user interface. An icon is selected that will have a number of associated instructions. However, all that will be noticed is the result. The Application layer takes requests and gives output to the user in the user's required form. This is where the network services the application program when using an agreed upon protocol (such as HTTP, IMAP, or even Telnet)

The Tortuous Path

Applications utilities such as File Transfer, Access Control, and Management (FTAM) are programs that interface to the Applications layer. If you wanted the external program to review a file located on some distant (but accessible) point on the network, it would, of course, have the file name but probably not its location. This desired information would be passed to the Applications layer—it is the original (for this particular application) data to be transmitted. The request is a file query regarding location and path. The request would go to the Presentation layer. There additional information and directions about translating the Applications data into a standard format would be tacked on before the Application data is added. The additional information will be needed by the receiving Session layer before it passes it the Applications data.

Next, the request would go to the Session layer, which would queue it for execution, establishing a virtual circuit (that will need to be disconnected when the transmission is completed). Again, this additional information is tacked in front of the added Presentation layer information (which is in front of the Applications data, which is in front of the actual file request).

Next, the Session layer passes the request on to the Transport Layer, where it is packetized (a better word here than "framed" because framing is actually done in the Data Link layer) into a packet data unit (PDU). Depending on system constraints there will be a minimum and a maximum packet size in bytes or octets (either one means 8 bits). The Transport layer is aware of the system's maximum packet size, and if the data (not a file request, as we are considering here) exceeds that length, Transport will have to determine how many packets will make up the sessions. At the Transport layer the data is broken up and sequenced (provided, of course, that you are using a connectionless transmission).

Layer 3 knows the destination (it is in a memory location and put there by the packet request data and may very well be the router address); will determine the routing path and add the routing information in front of the Transport layer's information. Layer 2 "frames" all the data, adds start and stop delimiters, generates and adds the FCS (Frame Check Sequence or CRCC is an error-detection feature). Finally, Layer 1 accesses the media (as dictated by Layer 2 Media Access Control) and dumps this data out onto the media in accordance with established rules. At the receiving end user, the same things happen only in reverse. At each step up, the bits added to the original data are stripped away for that layer's use, until finally only the original data is presented to the user at the receiving end. Figure 2-3 illustrates the encapsulation of each layer as it proceeds from the Application to the Physical. The receiving end is just the opposite of figure 2-3, with each layer stripping off its information and using it to pass information up to the next layer.

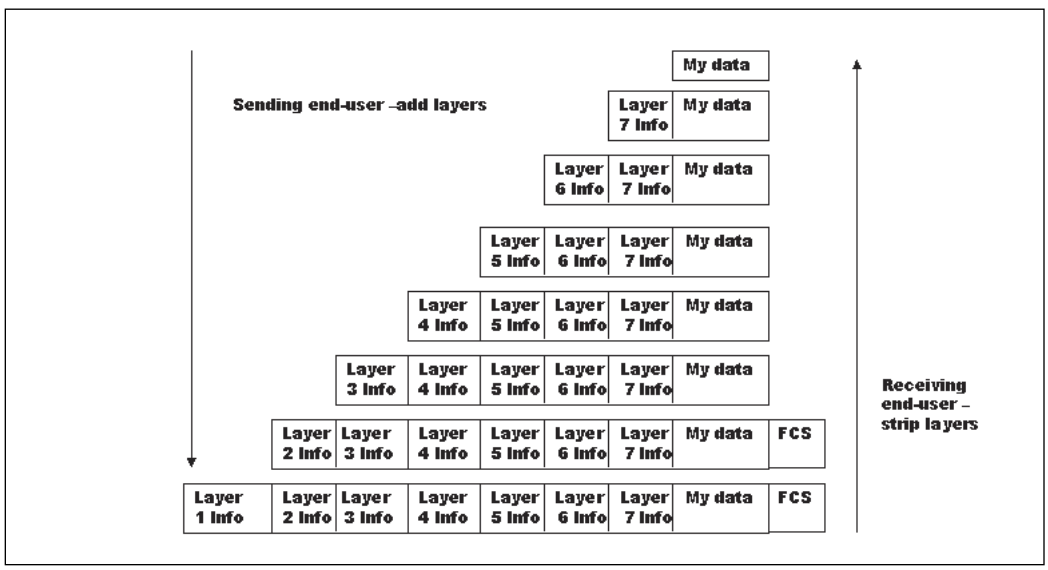


Figure 2-3. Layer Encapsulation

The Internet Model

Originating about the same time as the OSI/ISO model, the Internet has defined data communications into only five (5) layers, as figure 2-3a illustrates.

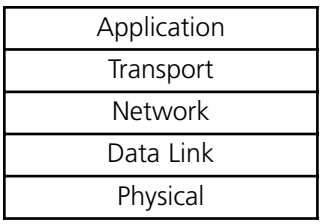


Figure 2-3a. Internet Layers

The Internet model is the most widely used protocol model for all data communications. The Data Link and Physical layers generally follow the IEEE 802 model described next, but the Network layer always uses the IP (Internet Protocol) format, which is currently defined as either version 4 or version 6. These versions are defined by Internet standards that can be found at the following web pages:

IP version 4: http://www.ietf.org/rfc/rfc0791.txt
IP version 6: http://www.ietf.org/rfc/rfc2460.txt

It is expected that within the next few years, IP version 6 will replace IP version 4. However, since version 4 is in such wide use, the two standards are likely to be in parallel operation for many years.

The most common Transport protocols are TCP and UDP (User Datagram Protocol). UDP is a simple protocol that requires no acknowledgement or error checking, while TCP uses a similar format to UDP data frames but overlays acknowledgement and end-to-end error checking to guarantee delivery of all data.

The TCP/IP suite now (as currently delivered) also provides the following standard applications, in addition to the transfer of data: (While some of these are true application layers, most reach into the presentation and session layer. The TCP/IP suite makes no claim to be OSI compatible.)

SMTP	Simple Mail Transport Protocol
SNMP	Simple Network Management Protocol
FTP	File Transfer Protocol
NTP	Network Time Protocol
DHCP	Dynamic Host Configuration Protocol
LDAP	Lightweight Directory Access Protocol
MIME	Multipurpose Internet Mail Extensions

SOAP	Simple Object Access Protocol
CIFS	Common Internet File System
Modbus-IP	A recently added data transfer command sequence protocol

The IEEE 802 Model

IEEE 802 was established as a local area network specification. This standard divides the (OSI) Data Link layer into two distinct sublayers: the Media Access Control (MAC) and the Logical Link Control (LLC) (see figure 2-4). Various SAPs are defined for the sublayers' connectivity. IEEE 802 also defines how different standard networks are to be connected (physically), what form of media access they should take, and how you interface your user data to them (Data Link layer). The number 802 (February 1980 was the first meeting of this committee) is the designation for the main IEEE committee for LAN standardization and specification. There were and are various subcommittees refining portions of the 802 specification (which defines many different aspects of LAN technology). An example is the 802.3, the standard for a Carrier Sense, Multiple Access/Collision Detection (CSMA/CD) network, a formalization of Ethernet. The standard 802.3 originally had two physical specifications, but it has many more now, as will be explained later. The 802 model is illustrated in figure 2-4.

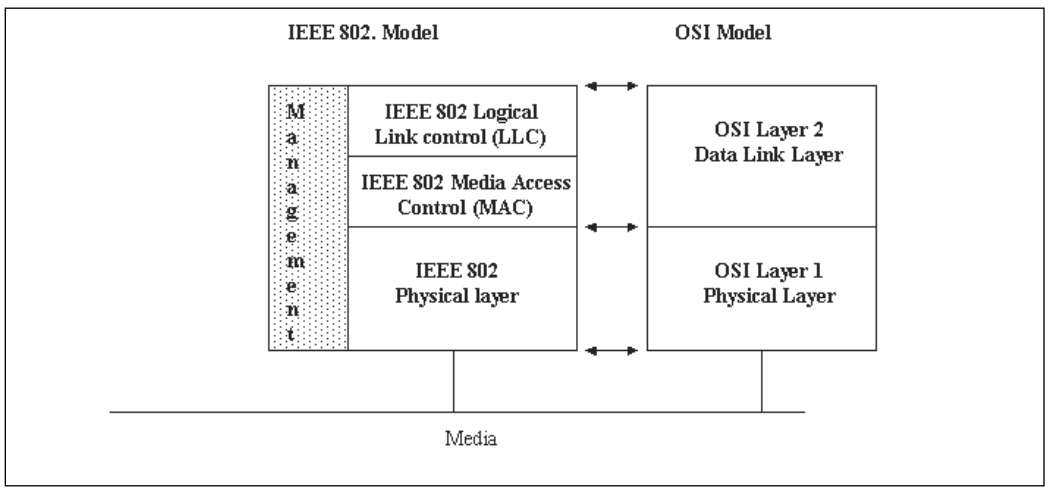


Figure 2-4. IEEE 802 Model

IEEE 802 specifies seven different LAN technologies, many of which will be discussed at some length in later chapters:

IEEE - 802.3 CSMA/CD Network (—Now known as Ethernet) and describes (at present) six LAN types: 10BASE5, 10BASE2, 10BASE T, 100BASE T, 1000BASE T, and 10000BASE T.

IEEE - 802.4 Token Passing Bus, a bus topology that uses a token passing access. Three different types are described, two broadband and one carrier band. Token Passing Bus was the basis for the now obsolete MAP/TOP effort.

IEEE - 802.5	Basically describes a nonproprietary version of the IBM™ Token Ring, running at 4/16/100 Mbps.
IEEE - 802.7	Broadband networks (essentially DSL and cable modems)
IEEE - 802.11	Wireless local area networks.
IEEE - 802.15	Personal area networks.
IEEE - 802.16	Wireless wide area networks.

The rationale behind the IEEE 802 model and its specifications is that if you meet the external interface requirements of the Logical Link Layer (LLC), then your communications will work regardless of the underlying MAC technology being used.

Unless a Layer 3 (Network) is provided to the LLC, the system is not routable outside of its own network. The reason for this is that there is no provision or mechanism for finding addresses other than data link (Layer 2) address, and hence the system is not routable to anywhere off of this one network. Many local area network protocols were not designed to be routable (such as Microsoft's NETBeui and IBM's SNA) by themselves. It should probably be noted here that many older industrial LANs (DCS/PLC systems) are "islands of automation" and were not designed to be connected to any other system. Originally, most industrial networks did not employ routable protocols. This was because there was no requirement to route because their network topologies were 'flat' (no computer had to send a message via an intermediate computer to reach any other computer on the network), so that overhead could be minimized, as could all of the other layers with the exception of Layer 1, Layer 2, and Layer 7. Figure 2-5 illustrates a typical industrial node. It should be stated here unequivocally that each of these protocols is a pattern of 1s and 0s. As a result, this pattern may be encapsulated into a routable protocol (as user data) and sent over many networks to a final destination (running that protocol), with the only necessity being spoofing time-outs.

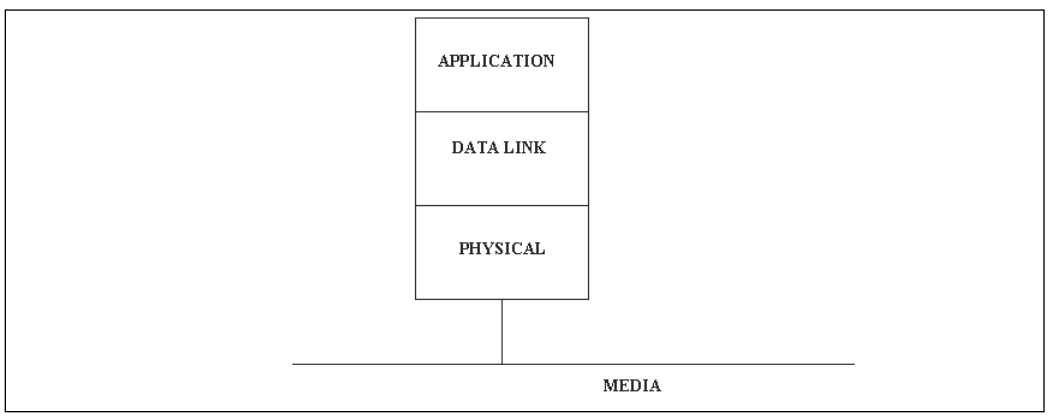


Figure 2-5. Typical Industrial Network Node

Application Models

One other set of models remains to be discussed before we move on to the details of each model. These are application models—that is, how communications and networking software is applied. We will look at four of them—one-tier, two-tier, three-tier, and N-tier—but we will begin by looking at the Application model itself (see figure 2-6). Note the three services: User, Business, and Data. These will be more fully explained in later chapters.

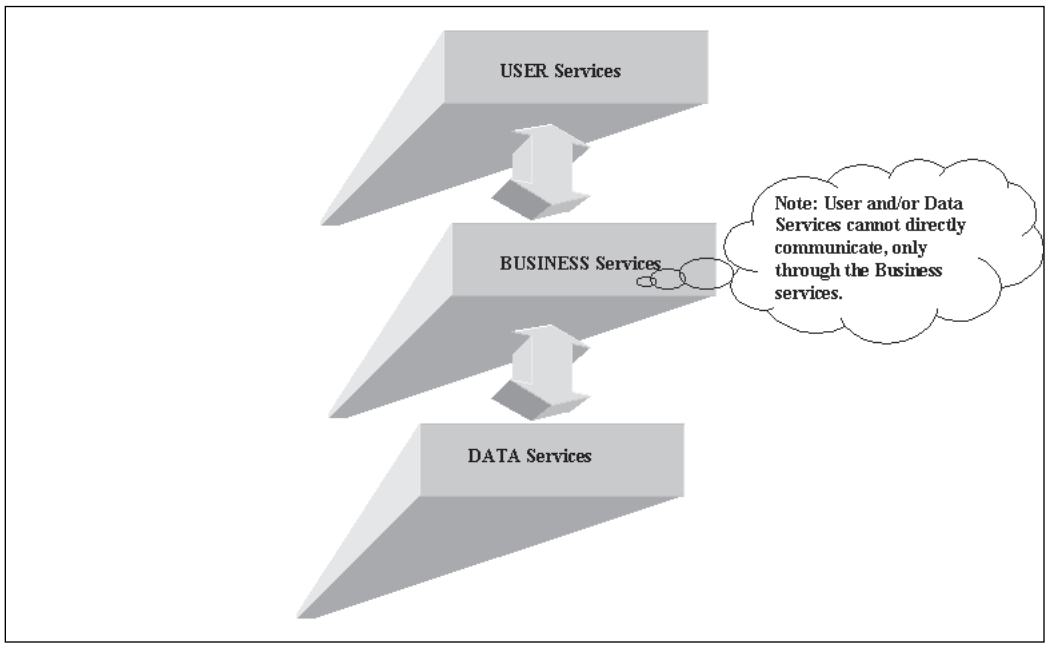


Figure 2-6. Applications Model

User services are the graphical user interface and presentation logics. Data is the ones and zeros we wish to organize and process. Typically, we place the data organization in a database system. The database will have a management system, and that is the name typically given to the database program. The majority of databases of any size are managed by relational database managers (such as Access, Oracle, SQL Server, etc.). Though the data doesn't have to reside in a database, in most cases it does. The Business services are an extremely important part of the system; they supply the organization and the boundaries, procedures, and rules.

Where these three services reside determines which tier model you are using. If you have a standalone machine (say a PC running a check-balancing program in which check data is stored in a personal database such as Access) then the User services are your graphical user interface (GUI), the rules are in the check program, and the Data services are provided by Access, all on one machine—hence, the name one-tier. A *one-tier* model is shown in figure 2-7.

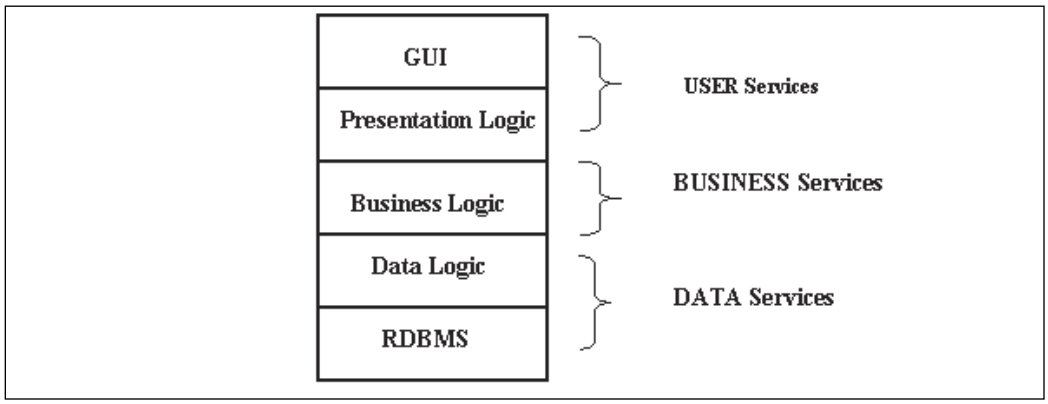


Figure 2-7. One-Tier System

Two-tier systems are typically called client-server systems. A server is any device that shares its resources; a client is generally an end user, typically a workstation that uses resources.

Client-server systems came about in the 1980s when personal computers were networked. Client-server refers to the functional model for the relationship between the node and the server. The client-server software architecture is query-response based (and modular in its infrastructure, presentation, business, and data services). It is intended to improve network performance in comparison to a centralized server-based architecture that uses time-shared access.

A client is defined as a user of services, and a server is defined as the resource for services. A single machine can be both a client and a server depending on the software installed on it and its configuration, as described previously for the one-tier system.

The original PC networks were based on file sharing in which the server (the provider of resources) downloaded files from a shared location to the workstation (requester of services). The application is then run on the workstation. This arrangement will work best if the number of stations requesting shares and the amount of data moved are small.

Because of the performance limitations of file sharing, the client/server arrangement gained popularity—that is, as memory and data storage increased in capacity and dropped in per bit price. The reduced system costs for microprocessor-based systems (for servers in particular) and the appearance of powerful relational database systems enhanced the popularity of the database server application, and it replaced the simple file server. Using a relational database management system (DBMS), user queries could be more efficiently handled. By using a query-response system the DBMS significantly reduced network traffic (by requiring only a query response rather than a file transfer).

In a two-tier architecture, the interface to the presentation system is located in the client workstation, and the DBMS is in a server that typically handles multiple requests from mul-

tiple clients. Where the Business services are located determines whether this is a "thin" or a "fat" client. If the Business services are located in the client it is a fat client; if they are located in the server it will be a thin client. Typically, if you use a browser you are a thin client, and if you use an application program for the Business services (for Windows it will be with a WIN32 interface) you are a fat client.

As well as it performs vis-à-vis the original file-sharing scheme, the two-tier approach does have its limitations. Because each user maintains a connection while using the DBMS, the number of possible simultaneous connections is limited. This user connection is maintained even when there is no data transfer (and actually until the client logs off, which in some cases could be eight hours or more).

In summary, we have two types of clients—thin clients and fat clients—based on where the Business services reside. You could actually run both the server and the workstation on one computer and have two tiers on one machine. Typically, however, a server is a separate machine. The server's requirements will depend upon whether it is a fat or thin client. Fat clients require higher-power workstations, but the server's demands are only for the database application. Thin clients require a very robust server as both the Business and Data services for each client run on the server. Figure 2-8 illustrates a two-tier system.

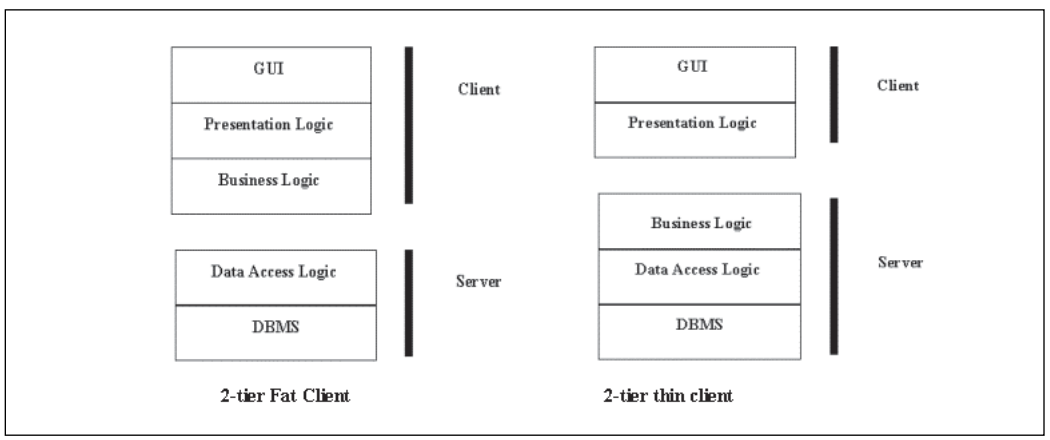


Figure 2-8. Two-Tier System

Three-tier and n-tier systems use application servers and very thin clients. A three-tier system is typical of a distributed network (intranet, industrial network, etc.) in which there is an application server running the Business services. N-tier describes Internet connectivity. Three-tier and n-tier systems are illustrated in figures 2-9 and 2-10, respectively.

The three-tier architecture was designed to avoid the limitations of the two-tier system. In the three-tier system, a middle tier was added between the presentation interface at the client and the interface to the DBMS Data services. By placing the Business services (or

rules) in the middle tier and generally on its own application server, it can perform queues, run applications, perform transaction processing, prioritize, and schedule.

The three-tier application runs the Business services application on a host (application server) rather than in either the Data services server or the client system. This application server shares business logic, computations, and a data retrieval service (an interface to the DBMS). As a result, a number of efficiencies are realized: upgrades to the business rules need only be performed on the server; the interface to the database server has greater integrity; and so on. Because security and program integrity are on one machine, not many, administrative control is greatly facilitated.

The three-tier architecture offers significantly better performance with a greater number of clients than is possible with the two-tier architectures. This is because the DBMS is not held hostage to the users logged on. Typically, the user requests data (which is located in the DBMS), so the business rules determine if they are allowed and then interface the request to the DBMS. A response to the client is made in the form (nowadays) of a page (using a browser as the presentation logic). There is no permanent connection. After the response is obtained and the information is downloaded to the presentation page the connection is broken. It is reestablished when the client performs some other request.

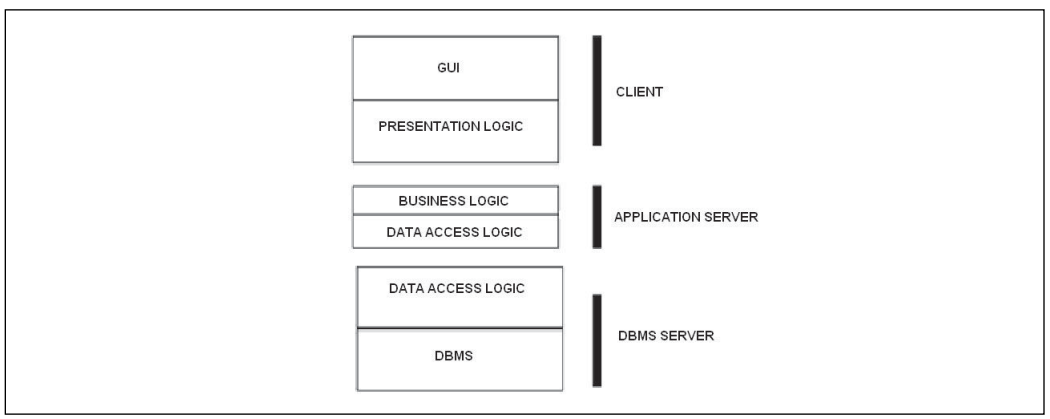


Figure 2-9. Three-Tier System

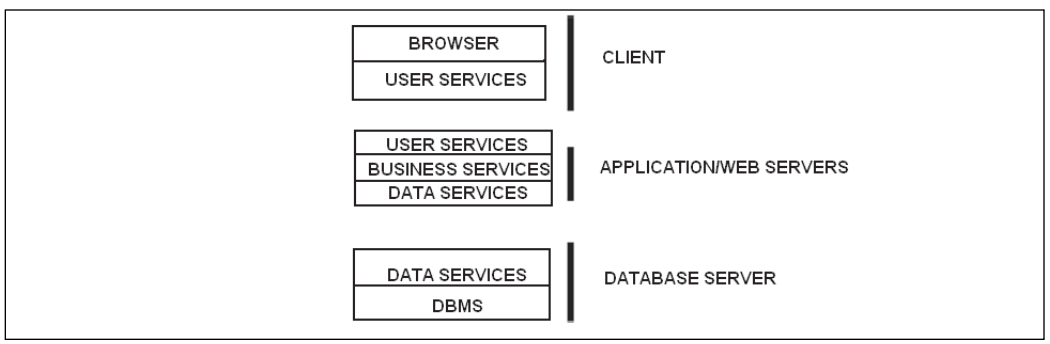


Figure 2-10. N-Tier System

Other terms involving the client-server, n-tier models abound. Two of these, producer-consumer and publisher-subscriber, basically describe how an application obtains information from a process that has undergone changes.

Producer-Consumer

An alternative method for exchanging data is the producer-consumer model. The server is the *producer*. It broadcasts data on the control system network or multicasts it to a multicast group. Clients are the *consumer* and listen for incoming data. Much of the data in a control system is of a read-only nature, and certain process values are of interest to many applications. If these values were supplied by a server, it would have to supply this data to the clients in a query-response mode. By not using a broadcast or multicast, the query response would waste bandwidth on the server's network and CPU cycles. In many applications, multicast would be more efficient. If multicasts are to be used throughout the control network, then all affected routers need to support them. Clients that are not on this control network or are not a part of a multicast group will have to acquire the data via other means.

Publisher-Subscriber

In control systems a better approach to the producer-consumer model would be to use publisher-subscriber as a method for exchanging data. In it, a client (*the subscriber*) communicates its query to a server (*the publisher*) and does not wait for a response. The subscriber does expect to receive a response within a designated period. This response can be for a one-time receipt of data or it can be either a request for data at regular intervals or data when values change. When working according to this model the server maintains a list of clients and the data in which they are interested. If a large number of clients want the same data, the server must acquire this data set only once and transmit it to clients on its list. This is quite efficient, much more so than the query-response (client-server) model.

A dedicated application will normally be the publisher. All other applications that depend on changes in the publisher application are called subscribers. As stated, the publisher maintains a list of the current subscriber applications. When an application wishes to be a subscriber, it will use a subscriber application provided by the publisher (just as in everyday life). Another application is provided to unsubscribe. Whenever the state or states of the publisher application is caused to change, the publisher will notify all current subscriber applications. The subscriber applications can then obtain the changed data at a convenient (for them) point in time. As long as the subscriber can process the available data within the constraints of application timing, the system will be real time.

Summary

This chapter briefly discussed the seven-layer model of the ISO OSI model for interconnection. Each layer is a set of functions that must necessarily be performed for an end-user-to-end-user communication regardless of media, system, or complexity. By defining the layers, standardization across vendors can be accomplished. It should also be

pointed out that just because two systems are OSI compliant does not mean they can communicate with each other. What it does mean is that if two or more systems use the same standards in achieving the same layers functions, then they will communicate.

The OSI model is used throughout this text (with explanations) to illustrate how the various layers are implemented by different standards or how whatever hardware or software being used is providing this function. This is after all a book about industrial data communications, an area that is rapidly becoming standards based. "Standards based" simply means nonproprietary or "open," a condition that should be welcomed by users and vendors alike.

We also looked at the IEEE LAN model and the main concepts behind it. The IEEE LAN model was intended to ensure communications provided you met the LLC external interface, regardless (within reason) of the underlying technology. Finally, we discussed a business Applications model and the differences between one-, two-, three-, and N-tier architectures. We also briefly touched on the alternate data exchange methods: producer-consumer and publisher-subscriber.

With this and the preceding chapters behind us we can now delve into the details of just how the various models, protocols, and communications are implemented.

Bibliography

Carnegie-Mellon University, Software Engineering Institute. *Client/Server Software Architectures—An Overview*. Pittsburgh: Carnegie-Mellon University, 2005.

Henshall, J., and S. Shaw. *OSI Explained: End-to-End Computer Communication Standards.* Chichester, UK: Horwood Ltd., 1988.

Martin, James. *Telecommunications and the Computer.* Upper Saddle River, NJ: Prentice Hall, 1990

Morneau, Keith. *MCSD Guide to Microsoft Solution Architectures.* Boston: Course Technology, 1999.

Stallings, William. *Local and Metropolitan Networks*. Upper Saddle River, NJ: Prentice Hall, 2000.

Stiefelmeyer, George. (as quoted in various lectures) Great Falls, VA: Stiefelmeyer International Limited, 1994

Thompson, Lawrence. *Industrial Data Communications.* 3d ed. Research Triangle Park, NC: ISA, 2001.

3
Serial Communication Standards

We've already introduced the concept of serial communications, the placing of one bit after another on a single media channel. It is the most prevalent form of data communications. However, there are many differences between transmitting a text file by modem to a bulletin board and sending data to the server over a 100-Mbps network. Both are serial, but they differ in many ways. This chapter focuses on three EIA/TIA serial standards for 232, 422, and 485 as well as the ancillary 423 and 530 standards. Since the United States now sits on the international standards committees, the EIA/TIA standards have their equivalency in ISO standards, and indeed most have been changed to meet the ISO standards. Serial interface to the PC is being accomplished by newer and much faster technologies, so we will also focus on four PC-based standards, USB 2.0, Firewire (IEEE 1394), SATA, and PCIe, that will impact industrial use and applications.

Definitions

Understanding how data circuit-terminating equipment (DCE) and data terminal equipment (DTE) are defined is essential to any discussion of data communications. From the hardware device dominant in the 1960s to today's software interfaces these terms are used throughout data communications applications (and not always correctly).

Data termination equipment (DTE) means the end device (either data source or destination) that originates data to or receives data from a piece of data communications equipment (a printer, computer, multiplexer), where the data transmitted is native (that is, as stored and operated on the equipment) and intended for transmission to another device. Figure 3-1 illustrates a DTE.

Data circuit-terminating equipment (DCE) is the point that is nearest the communications line external to the equipment. Examples are modems, line drivers, multiplexer composite output, and so on. Usually, the DCE is connected to a DTE, forming a complete data station. The DTE is farthest from the communications line; the DCE is the closest to the outside line. Figure 3-1 illustrates a typical DCE-to-DTE arrangement.

EIA/TIA Standards

Generally, the EIA/TIA serial standards present the electrical and/or the mechanical interface between the DCE and the DTE. Typically, these standards are for the Physical layer only and make no allusion to data link schemes or any of the data link control protocols. An analog DCE

such as a modem is concerned with the line modulation type, media access, line speed, and so on. These standards only deal with the digital link between DTEs, DCEs, and other devices.

Figure 3-1 illustrates a typical DCE/DTE setup and the signal types found at various points. Any discussion of the serial transmission digital data interface between DTE and DCE starts with one of the most widely used digital standards ever known: the Electronic Industries Association (EIA) 232, which is now in its F version. This standard is frequently called by its original name RS-232 (RS means "recommended standard").

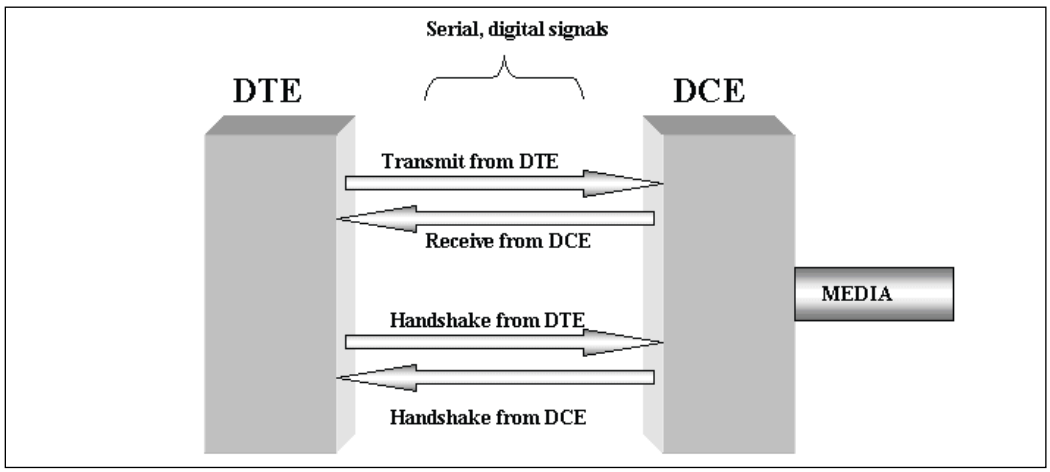


Figure 3-1. DTE to DCE

Example 3-1:
Problem: You have a PC with an external modem. Obviously, the modem is the DCE, and the computer is the DTE. If the computer operates a printer over a serial line and the printer is a DTE, what is the computer?

Solution: Though it would appear logical to say that the computer is the printer's DCE, it is in fact a DTE as well. To actually connect a serial printer to a computer you need a null modem. A null modem is a male-to-female connector in which the data lines are crossed pins 2 to pins 3 (and sometimes the handshake lines as well). The null modem would provide the solution to connecting a serial printer (DTE) to the serial port of the computer (DTE).

TIA/EIA 232(F)

TIA is the Telecommunications Industries Association. Its 232(F) standard is for the interface that makes possible the connection between DTE (which could be a controller, printer, or computer) and DCE (modem, etc.) by employing Serial Binary Data Interchange. When this

standard was first conceived, the DTE was either a teletypewriter or a dumb terminal (a terminal with only data handling, no peripherals of its own), and the distant end DTE was probably a computer. The then RS-232(C) was originally intended to describe this interface; it was never anticipated that it would be used by everything from calculators to multiplexers.

Up through RS-232(C) (1968), the standard specified only that there be a 25-pin connector (no requirement for a male, female, or subminiature type D connector) and only described the 25 pins and their actions. The standard dictated a maximum transmission distance of fifty feet. An addendum to the standard (1972) stated that the 232 standard was obsolete, should not be used in new designs, and should be limited in existing designs to 9600 bps. The addendum further stated that EIA RS-449, accompanied by electrical standards EIA 422 or 423, should be used instead. EIA 449 apparently never really caught on, however, and RS-232 continued to be used, even at 19.2 Kbps (or higher for non-standard applications).

The "D" version of this standard, made standard in 1986 and published in January 1987, changed several items. The E version brought TIA/EIA 232 in line with ITU V.24 and V.28 and ISO 2110. It specifies the 25-pin connector dimensionally and electrically. It includes several new circuits and redefines the protective ground. The following terminologies were affected:

- DCE, which was "data communicating equipment" or "data-set," is now "*Data Circuit-terminating Equipment*."
- Driver" is now "*Generator*."
- Terminator" is now "*Receiver*."

When microcomputers began replacing dumb terminals, the connector type used not only depended upon the connector's physical gender, but on the programming of the associated universal asynchronous receiver-transmitter (UART—an integrated circuit that performed serial/parallel conversion with associated control and logic functions).

The reason for the subminiature D type is simple: it was the connector used on the Bell modems in the late 1960s. The DCE was a female connector, so it (logically) was assumed that the DTE was a male connector. Because the connector type was not explicitly stated and because the DCE or DTE function could be programmed into many microcomputers, one was never sure what type of cable connectors were needed or how many lines would be in the cable.

An entire industry segment grew around this uncertainty, manufacturing items such as "gender menders" (two connectors, both either female or male, that were connected back to back), which caused the cable end plug to change its gender. Also produced were "null modems" (a male and a female connector connected back to back, with lines 2 and 3 crossed and perhaps pairs 4 and 5 and 6 and 20 as well) and "breakout boxes." A breakout box typically has a male and a female connector and, lying between the connectors, has

some or all of the twenty-five lines, which are brought to test points that have switches to break the normal path. You would then jumper different lines until the desired result was obtained. Several enterprising companies made breakout boxes complete with microprocessors that could indicate how to correctly wire the connections so the circuits would function.

TIA/EIA 232(F) is the current (when this book went to press) version. It is basically the same as the E version, but brought into line with international standards. The main difference lies in some electrical specifications involving signal rise time. (You had to buy an extra standard to acquire the 9-pin specifications). Table 3-1 provides a list of pin terminations and circuit numbers for the 25-pin connector (F version). Table 3-2 lists the 9-pin connectors' terminations.

Eq. Ckt	CCITT	Pin	Function
AA	101	1	Protective Ground (Frame Ground)
AB	102	7	Signal Ground—all signals are referenced to pin 7
BA	103	2	TxD Transmitted Data, DTE to DCE
BB	104	3	RxD Received Data, DCE to DTE
CA	105	4	RTS Request to Send, DTE to DCE
CB	106	5	CTS Clear to Send, DCE to DTE
CC	107	6	DCE Ready, DCE to DTE
CD	108.2	20	DTE Ready, DTE to DCE
CE	125	22	Received Line Signal Detector (Ring Indicator DCE to DTE)
CF	109	8	Data Carrier Detect, DCE to DTE
CH	111	23	Data Rate Selector, DTE to DCEs
CI	112	11	Data Rate Selector, DCE to DTE
DA	113	24	External Transmitter Clock, DTE to DCE
DB	114	15	Transmitter Clock, DCE to DTE
DD	115	17	Receiver Clock, DCE to DTE
RL/CG	140	21	Remote Loop Back DTE to DCE
LL	141	18	Local Loop Back, DTE to DCE
TM	142	25	Test Mode, DCE to DTE
			Pins of Historical Importance
SBA	118	14	Secondary Transmitted Data
SBB	119	16	Secondary Received Data
SCA	120	19	Secondary RTS
SCB	121	13	Secondary CTS
SCF	122	12	Secondary Data Carrier Detect

Table 3-1. TIA/EIA 232(F) Pin-Out

Eq. Ckt	CCITT	Pin	Function
CF	109	1	**DCD** Data Carrier Detect
BB	104	2	**RxD** Receive Data
BA	103	3	**TxD** Transmit Data
CD	108	4	**DTR** DTE Ready
AB	102	5	Signal Ground
CC	107	6	**DSR** DCE Ready
CA	105	7	**RTS** Request to Send
CB	106	8	**CTS** Clear to Send
CE	125	9	**RI** Received Line Signal Indicator

Table 3-2. DB-9 Pin-outs

For the equivalent circuits, if the first letter is an A, it is a common circuit. If it is a B, it is a signal circuit; if it is a C, it's a control circuit, and if it's a D, it is a timing circuit. S indicates a secondary circuit. It's worth noting here that old terms are hard to change. Technically, the DTR signal is not "Data Terminal Ready," but "Data Termination Equipment Ready." Somehow it doesn't have quite the same ring to it. The DB (the D stands for Subminiature D - the meaning of B is not clear or defined - and these types of connecters are identified as DB) connectors are illustrated in figure 3-2.

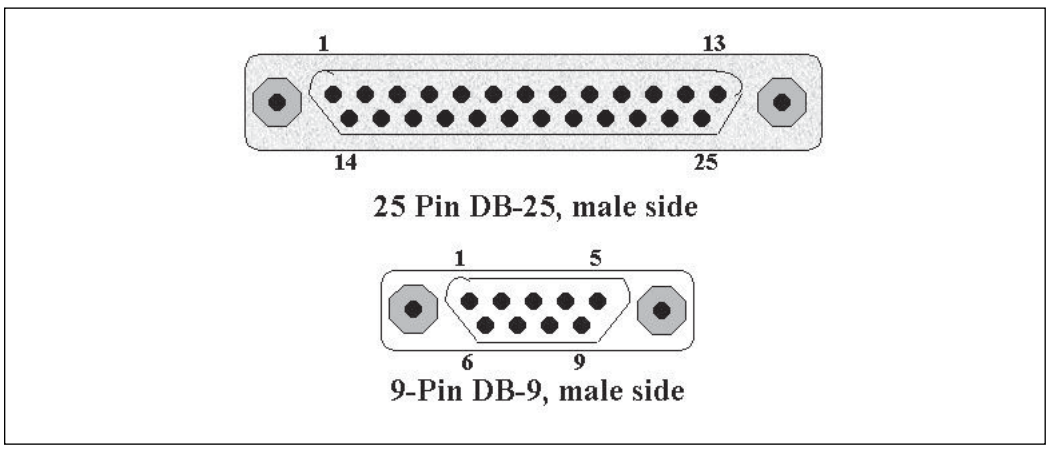


Figure 3-2. EIA/TIA 232 Connectors

The pins of historical importance were concerned with the Bell 202 modem and the reverse channel, also called the secondary channel. This low-speed part of the bandwidth was used only for ACKs and NAKs (responses to the reception of a packet without errors) to keep them from having to turn the line around, and thus losing time to modem setup. Some modems (at higher speeds) still exist that use a secondary (also called reverse channel)

either as a receive acknowledgment or as a transmit poll (clear to send). Other pins have different uses according to who is implementing the circuitry. Typically, only the DB9 pins, listed in table 3-2 by equivalent circuit and pin, find use in asynchronous systems today. Figure 3-3 illustrates the waveform voltage limits found in RS-232(C) and EIA 232(F). The teletypewriter origin of this signal is apparent insofar as the most positive signal condition is a logic zero (or SPACE), whereas the most negative signal is a logic one (or MARK). Teletypewriter circuits had long used an OFF condition (SPACE), with no current, and the ON condition (MARK), which was a negative voltage (in relation to earth ground). This arrangement is known as neutral signaling. Polar (for polarized) signaling does not have a legal no-current state. Instead, either a positive or a negative voltage was impressed across the line, so the negative condition as a MARK was just carried on. As almost all logic in contemporary use has as its most positive state the TRUE or logical one condition, this tends to cause a bit of confusion, particularly when you're trying to observe EIA 232 signals.

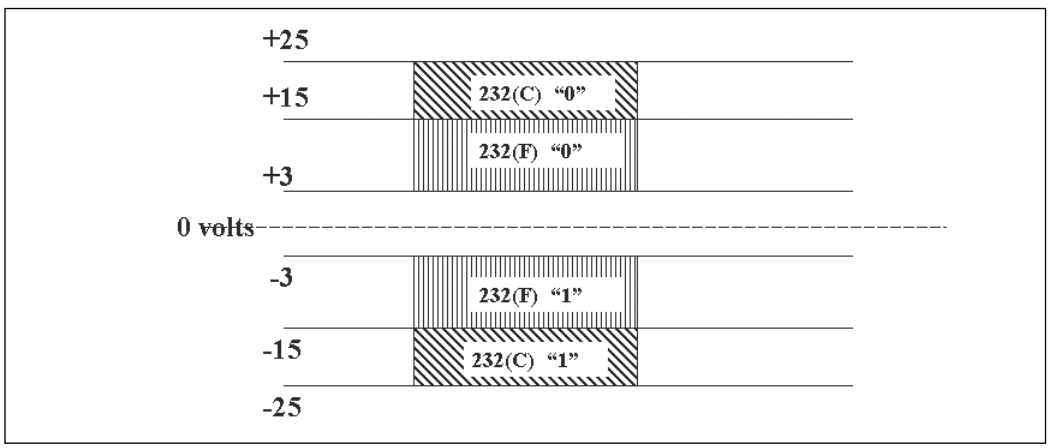


Figure 3-3. EIA/TIA 232 Voltage Levels

How did such a standard become so widely used? There was no alternative standard. The original standard was defining a very small application that over time blossomed into near universality. *Remember*, when a device says it is "232 compatible" that means only that it outputs a signal somewhere between ±3 to ±15 volts, that certain circuit input and output impedances are within a range, and, with luck, that the DTE (if identified) will have transmit data on pin 2 and expect to receive data on pin 3. Whether it uses pin 20 (DTR) or pin 4 (RTS) to provide a handshake is up to the manufacturer. Many modern data communications circuits, particularly at the higher speeds, use software handshaking and do not use the hardware flow control (handshake) pins—although most PCs expect these controls to be enabled before they will communicate. EIA 232 is specified out to 20 Kbps (although many an EIA 232 circuit is humming along at 56 Kbps, some much higher).

In 1972, EIA recommended a new standard for media connections that starts at 20 Kbps and is designed for even higher speed data. EIA 449 was to replace RS-232 at all speeds. It

had a 37-pin main connector, relegating the secondary channel connections to a separate plug. If it wasn't used (as was increasingly the case), it didn't even have to be connected.

EIA 449: Interface Standard

In contrast to RS-232, EIA 449 used the EIA 422 and EIA 423 standards for media and electrical descriptions. It described only the pins, definitions, connector specs, and functions, and referred to EIA 422 and EIA 423 whenever electrical connections to the media interface were described. EIA 449 was originally intended to phase out RS-232, but EIA 232 is now in the F version, and EIA 449 has been replaced by EIA 530 for circuits above 20 Kbps. It appears that EIA 232 will be around just a while longer.

EIA 422 and 423

EIA/TIA 232 specifies the pin-outs, line characteristics, input and output impedances, in fact, the entire interface. As an interface may have more than point-to-point circuits, the newer standards tend to describe just the interface's physical and synchronization characteristics. A number of standards were developed that defined the electrical interface and could then be referenced from the physical standard. The development of these new standards prevented duplication and contention over small changes in different written standards. Two electrical interface standards now in place are EIA/TIA 422 and EIA/TIA 423.

EIA/TIA 422: Balanced Interface

EIA/TIA 422 is an electrical-media standard, specifying input and output impedances, line lengths, rise and fall times, signaling speeds, voltage levels, and so on. The standard calls for a ±200-mV-to-±6-volt signal, with the most positive condition being the logical zero state.

EIA/TIA 422 is a "balanced-to-ground" specification. This means that both of the transmit terminals and both of the receive terminals have the same resistance to ground (i.e., they are balanced to ground as figure 3-4 illustrates). Being a balanced to ground interface, it does not have to contend with the charge/discharge of uneven line capacitances, which it would have had to if it were unbalanced. Noise is normally found in the common mode (between the 0VDC reference and the signal line), and the balanced line uses differential inputs and outputs, which by nature have a high common mode-rejection ratio so noise is reduced as a factor in signal quality. These factors (and some others) allow a balanced line to operate at a higher speed. EIA/TIA 422 is specified for 20 Kbps to 10 million bits per second (10 Mbps). All of the data provided in the standard is relative to a twisted pair (22 AWG).

Because the signal voltage is measured relative to one pin (usually A), the polarity of the output pins is important. Typically, any interface will mark the two pins as + and – (or perhaps A and B).

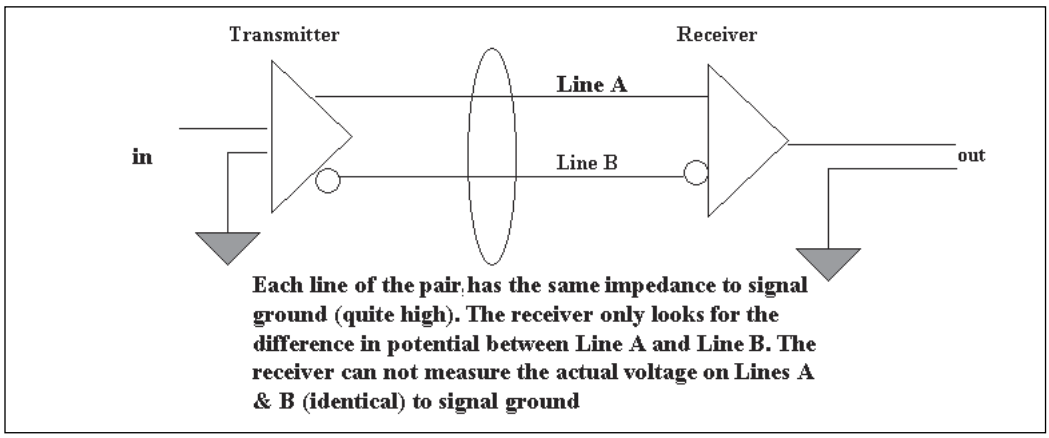


Figure 3-4. Balanced to Ground

Using a balanced system requires that you have two terminals, both with equal impedance to the signal common. A "1" condition is determined when A is negative with respect to B, and a SPACE is determined when A is positive with respect to B. The signal will have a minimum level of ±200 mV and a maximum of ±6 volts.

EIA 422, which is specified for one transmitter and up to sixteen receivers, is mostly used today for point-to-point transmission involving four wires (two pair) for duplex transmission. The one transmitter can drive sixteen devices, but the standard does not allow for contention. As a result, if you want a network that contains more than one transmitter you would have to ensure that the system would not allow more than one transmitter to be active at a time.

EIA/TIA 423: Unbalanced Interface

EIA/TIA 423, on the other hand, is specified as unbalanced to ground. That is, the return for both transmit and receive is at a common potential: the circuit 0VDC reference point, which is referred to as signal ground or signal common. Figure 3-5 illustrates an unbalanced system.

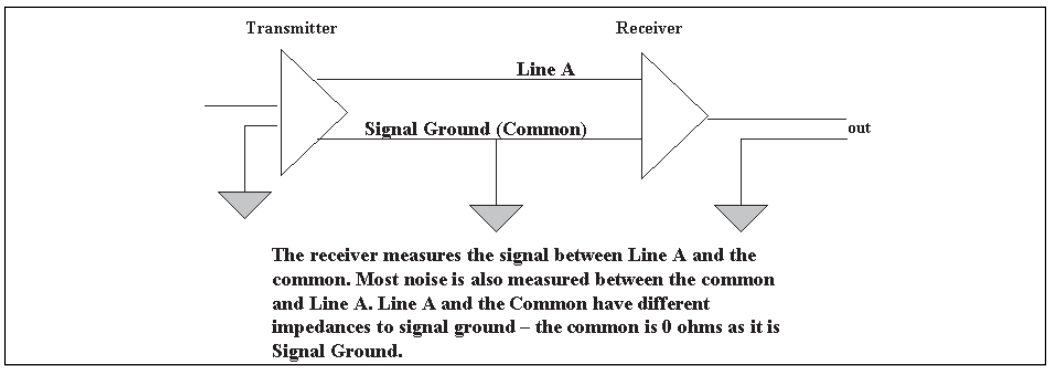


Figure 3-5. Unbalanced to Ground

The unbalanced system will run slower (for a given media and distance) than the balanced line. This is primarily because of the different charge/discharge times of the cable capacitance and the fact that noise is also referenced to the signal ground—meaning it is in the common mode. Actually, the 422/423 transmitters and receivers are identical! When the 422 is connected with one lead common to the signal return and a single "hot" lead, there is no discernible difference in operation below 20 Kbps. EIA/TIA 423 is rated at 20 Kbps maximum, as is EIA 232, which is also unbalanced to ground.

Note that EIA/TIA 422 refers to a normalized twisted pair. EIA/TIA 232 is an unbalanced line standard, and the EIA/TIA 423 receiver should be able to accept the EIA 232 signal. It is entirely probable that both balanced and unbalanced lines may be found in the same cable: data and timing signals (because of the high line rate) will be balanced lines, while control lines may very well be unbalanced.

There are, of course, those recommended maximum distances, which these standards indicate are the maximum lengths (though they are usually not given for the maximum data rate) before the signal becomes degraded. Most specified distances are shorter than those found in actual use. Whereas EIA/TIA 232(C) specified a maximum length for all data rates (50 feet), EIA/TIA 232 (D) had a sliding distance scale based on speed. Meanwhile, EIA/TIA 232 (E and F) has no distance limit but uses the line capacitance as the limiting factor. With modern unshielded cable this could extend easily to 40 meters. EIA/TIA 422 and EIA/TIA 423 specify distance in terms of the data rate.

At 10 Mbps (EIA/TIA 422), the distance between a transmitter and receiver is limited to 3 meters (10 feet). This is rather unfortunate because unshielded twisted-pair (UTP) cables used in Ethernet (10BaseT) operate at 100 meters (390 feet) at 10 Mbps, and 100BaseT will operate at 100 Mbps over that same 100 meters. Both use balanced pairs. EIA/TIA 422 and EIA/TIA 423 do not specify protocol, timing sequence, quality limits, or pin assignments.

EIA/TIA 485(A)

The standards described in the preceding sections had, as their general concept, point-to-point communications. In this section we describe multipoint communications, the architecture that has more than two stations, as an adjunct to most of the previous standards.

EIA/TIA 485 was written to describe a transmitter-receiver combination that is capable of multipoint operation. In fact, this standard will allow any combination of up to thirty-two transmitters (generators) and receivers on the same two-wire line. It specifies a balanced line for data transmission and reception, much as does EIA 422. But where EIA 422 allows only one transmitter and specifies that all other devices must be receivers, EIA 485 uses tri-state logic and ties the transmitter and receiver to the same wire pair. This is a bus arrangement, so the addressing must be taken care of in software, as must the response to all commands, and so on.

The major differences between EIA 422 and EIA 485 are detailed in table 3-2.

Function	EIA 422	EIA 485
Minimum output voltage	200 mV into 100 ohms	1.5 V into 60 ohms
Current (short to ground)	150 mA maximum	
Current (short to + source)		250 mA peak
Rise Time	<10% bit time	<30% bit time into 54 ohms, 50 pF load.

Table 3-2. EIA 422/485 Differences

If the differences between EIA 422 and EIA 485 do not have apparently deep and lasting significance, one can at least appreciate the fact that the EIA 485 standard was designed from EIA 422 and that a circuit could be designed easily to accommodate both types of transmitter/receivers. The primary difference between the two is that EIA 485 can support contention (the situation in which two transmitters both become active at the same time in opposite polarity; if this is not supported a short circuit would result).

EIA 485 doesn't say what software is required to effect a serial multipoint network, timing requirements, protocol, or pin-outs. The typical pin-out is the subminiature 9-pin D connector, but it may be a "D" block, or any other termination that the manufacturer adopts. There must be some means of directing who shall speak, who shall listen, and in what order, and what to do if nobody is talking. These are the same issues that arise over and over regardless of the type of network used.

An EIA 485 network is probably one of the least expensive networks to implement with interface cards for PCs (they generally take the place of one of the COM ports), provided that no more than thirty-two points will be needed. The industry-standard 56 to 115 Kbps data rate is typically supported in commercial adapters. So it should be no surprise that this serial scheme is used in many commercial applications, including instrumentation systems.

Example 3-2:
Answer the following questions regarding an EIA 485 network.

1. Upon power-up, or initialization, who starts talking?
2. What method will give each station a fair amount of time, or priority?
3. Can one station talk to only one station, or to all stations?
4. What addressing scheme will be used?
5. If a station fails, will all be affected?
6. How do you add a new station?

These questions are actually answered by the Data Link layer, and in particular the Media Access Control. For EIA 485 they are provided (sold) by vendors or integrators. They are not answered by the standard (or the Physical layer).

EIA/TIA 530

This standard was intended to gradually phase out EIA 449 at the higher (above 20 Kbps) data rates. It specifies the EIA/TIA 422 balanced standard for its Category I circuits. Category II circuits may be unbalanced EIA/TIA 423 types. The main function of the EIA/TIA 530 standard is to complement EIA/TIA 232(E) above 20 Kbps (the upper limit of the EIA 423 standard receiver). The specified 25-pin connector is similar to the one described in our section on EIA/TIA 232, but the pin-out is much different. It is doubtful, however, that you will encounter this standard in today's market.

The functions of the named pins are much the same as in RS-232. Table 3-3 illustrates the differences between RS-232C, EIA 232E, and EIA 530. Note that EIA 530 (-) pins are same as EIA 232(E) pins.

Signal	RS-232(C)	EIA 232(E)	EIA 530
Frame Ground	1		
Shield Ground		1	1
Transmit Data	2	2	2(-) 14 (+)
Receive Data	3	3	3 (-) 16 (+)
RTS	4	4	4 (-) 19 (+)
CTS	5	5	5 (-) 13 (+)
Data Set Ready	6		
DCE Ready		6	6 (-) 22 (+)
Data Term Ready	20		
DTE Ready		20	20 (-) 23 (+)
Signal Ground	7	7	7
Carrier Detect	8		
Rec. Line Sig. Det		8	8 (-) 10 (+)
TX Timing DCE	15	15	15 (-) 12 (+)
RX Timing DCE	17	17	17 (-) 9 (+)
Local Loopback	18	18	18
Remote Loopback	21	21	21
Ext TX Clock	24		
TX Timing DTE		24	24 (-) 11 (+)
Test Mode	25	25	25

Table 3-3. Differences between RS-232(C), EIA 232(E), and EIA 530 Pin-Outs

Interface Signal Functions

A look at table 3-3 will show you that certain signals are common to all the interface functions. In the next few paragraphs we'll describe these "hardware handshake" signals when in normal operation. This knowledge is useful when determining whether a serial interface is operating correctly (or even will operate). The order in which the signals are described here is generic. However, when using modern devices you will encounter different sequences, and not all equipment uses all of these signals. It is quite possible to find a serial scanner that has only three lines, although five or six may be more typical. At any rate, the following sequence is what the PC normally expects at its serial port.

Ensuring Operability

DTE Ready—Data Terminal Equipment Ready. This must be TRUE (ON), which signifies that the Data Terminal Equipment (DTE) is ready to communicate.

DCE Ready—Data Circuit-terminating equipment Ready (formerly known as "DSR Data Set Ready"). This must be TRUE (ON). Sent from the Data Circuit-terminating Equipment (DCE) to the DTE, this signal determines whether the DCE is in the DATA mode and ready to communicate and (as used in EIA 232) whether the DCE is authorized to set up a link once it has detected ringing.

Setup Link

If you are using a dial-up (switched) circuit in which the DCE is not permanently connected to the transmission media use the following signals.

At originate DCE:
DCE Ready—Should be *TRUE*. This may be set at power-up, or upon RTS.
RTS Request to Send—Should be set *TRU*E by the DTE. At this point, the DCE dials up, establishes a connection, and so on.

At the answer DCE:
RI Ring Indicator—Receive Line Signal Indicator goes *TRUE*, which indicates that a call request is being made.
Upon receipt of RI, and after synchronization, the *DCE Ready TRUE* will be sent to the DTE. If you are using nonswitched lines RI is not used.

DCE synchronize—Both ends will have *DCE Ready as TRUE*. Originate has *RTS TRUE*. Then originate DCE will signal DTE with *CTS TRUE* to begin transmitting. As long as the answer DCE is receiving is *DTE Ready TRUE* then data will be transferred to the DTE.

Actual hardware flow control depends on only one set of the DCE/DTE Ready or RTS/CTS pairs. DTE/DCE Ready in IBM-based systems and RTS/CTS signals in DEC systems were used for handshaking. Contemporary transmission uses software rather than hardware hand-

shakes, and DCE/DTE Ready along with RTS/CTS are jumpered to provide UART actuation.

Signals required with software flow control
TD Transmit Data—Referenced to pin 7 (Signal Ground); the DTE outputs data to the DCE.
RD Receive Data—Referenced to pin 7 (Signal Ground); the DTE receives data from the DCE.
Signal Ground—The 0VDC reference (signal return).

Synchronous communication
Bit oriented—There are no start and stop bits; either the DTE or the DCE may supply transmitter clock (bit timing).

If the DTE supplies it:
(E)TC External Transmitter will be used to clock the transmit timing.

If the DCE supplies it:
TC Transmitter Signal Element Timing will be used to clock the transmit timing.

In either case:
RC Receiver Signal Element Timing will be used to clock out the received data.
Note that for asynchronous transmission that uses software flow control, only three wires are required for duplex transmission.

PC Serial Communications

Just as EIA 232 usage was greatly facilitated by the personal computer (PC), other buses internal and external to the computer have appeared. Some are actually networks themselves rather than just an extension of an internal bus. For a long time, external line speeds required only the EIA 232, then the EIA 422/485 connections or perhaps the SCSI extended bus. However, much higher speeds required a rethinking of the serial structure and the software that drives it.

Universal Serial Bus (USB)
The Universal Serial Bus was originally developed in 1995 by industry-leading companies. The concept was to define an external expansion bus that made adding peripherals to a PC as simple as plugging in a network jack. The design goals were low cost and true "plug-and-play" operation. Both were enabled by using an external expansion architecture. Figure 3-6 illustrates a laptop computer with a root hub showing connections to a USB hub, a printer, a mouse, a wireless access point, and a ready-to-connect a flash drive (currently available to 8 GB).

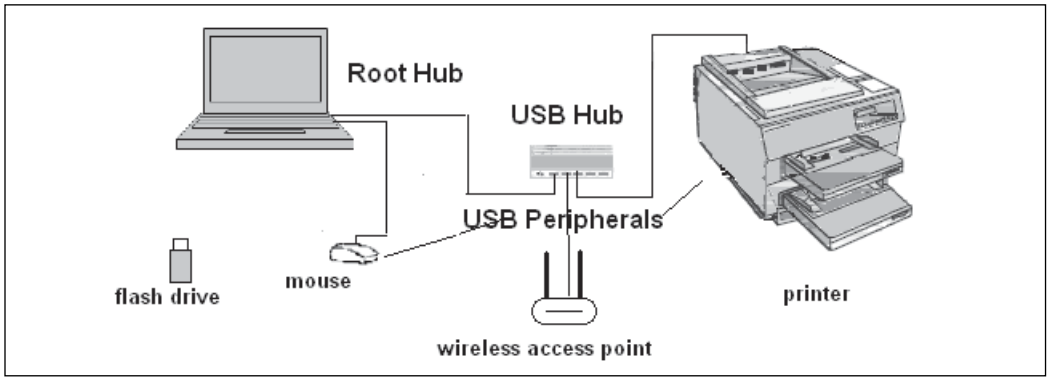


Figure 3-6. USB Connections

USB is currently in version 2.0 and runs at speeds approaching 480 Mbps. It is backward-compatible with the 1.1 version devices that run at either 12 Mbps or 1.5 Mbps. External hubs (counting the root hub) can be nested up to five deep, and USB 2.0 allows up to a maximum of 127 devices (including the root hub). Most PCs today come with two, four, or six (sometimes more) USB connectors.

One distinct advantage USB has over the older technologies is that it is truly plug-and-play. You may have to load a driver, of course, but if you use Windows XP or Windows Server 2003 (and, of course, the new Vista) you will find software support for all of the USB 2.0 specification. Additionally, USB 2.0 was modified to allow peer-to-peer operation. Therefore, a PC is not necessary. LINUX has had USB support for 2.0 since at least 2005.

The USB attachment cable may have either a Type A or a Type B connector. Type A is usually found on the PC, with Type B on the device. Figure 3-7 illustrates the connector types.

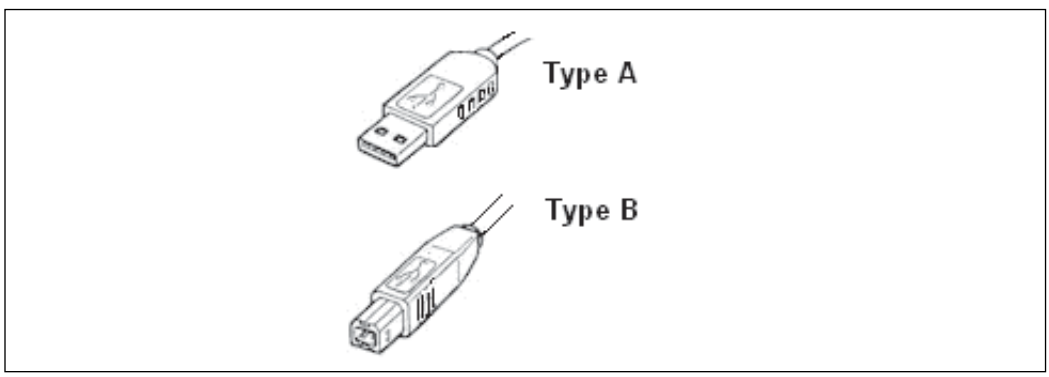


Figure 3-7. USB Connectors

There are a number of different "mini" USB Type B connectors, including mini 4 pin and mini 5 pin. However, we will focus only on the standard Type A and Type B, whose pin-outs are shown in figure 3-8.

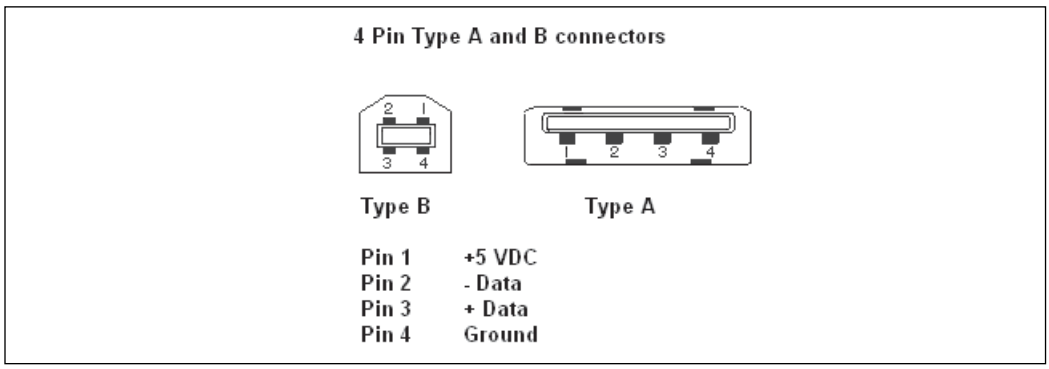


Figure 3-8. USB Type A and B Pin-outs

At the moment, USB, aside from PC workstations and their peripherals, has not achieved a significant presence in industrial settings. That said, however, there are areas where it begins to impact (both good and bad) industrial use, to wit: the flash drive (pen drive, USB drive, etc.). The flash drive has all but eliminated the use of floppy disks for data transfer. You may find flash-drive devices ranging from 32 Kbytes to (presently) 8 GB at affordable prices (and downright cheap if you look back a few years). The upside is that most maintenance technicians can now carry around with them the diagnostics and software for any number of uses (no more floppies—yay!). The downside is that flash drives make software and data theft much easier to accomplish.

IEEE-1394

IEEE-1394 is a fast *external bus* standard that supports data transfer rates of up to 400 Megabits per second (in 1394a) and 800 Megabits per second (in 1394b). Although the standard allows for rates of 1.6 and 3.2 Gigabits per second, there is (circa 2006) little product above 800 Mbps. Different manufacturers use the 1394 standard but do so under different names. Apple (the original developer) uses "*FireWire*," a trademarked name. Sony has trademarked *i.link*.

A single 1394 port can be used to connect up to sixty-three external devices in a daisy-chain connection in which each device is connected to the previous device. IEEE 1394 supports *isochronous data*—which means it delivers data at a guaranteed rate. This would make it ideal for industrial control devices that need to transfer data in real time. The IEEE 1394 protocol uses an 8B/10B signaling scheme, where 8-bit combinations are represented by 10-bit patterns. This minimizes the number of consecutive zeros allowed in a data stream. This scheme means, however, that the protocol is only 80 percent efficient in terms of the line data rate.

Though fast, IEEE 1394 is relatively expensive, particularly when compared to USB. Like USB, 1394 supports plug-and-play and hot plugging. Various cables are keyed so devices cannot be wrongly connected together. The original 1394 uses a 6-pin (supplying power) or 4-pin (no power) connector while the 1394b uses a 9-pin connector.

IEEE 1394 is a promising bus for industrial networks, and its adherents insist it is competitive with Ethernet. However, it does not at this time appear to be gaining any traction beyond the world of digital cameras and hard drives even though it would appear to make an ideal peer-to-peer transport. USB 2.0 has comparable speed, is also much less expensive, and appears to be ubiquitous. The marketplace will decide which technology is going to succeed in the industrial market.

SCSI Small Computer System Interface (Current Usage Only)
The SCSI interface has been around for almost two decades. It has a high data transfer rate and can be used as an external bus. It is, however, more expensive than USB, IEEE 1394, or SATA II (a very-high-speed serial interface for hard disks).

SCSI has moved in the past several years from a parallel configuration to a serial connection, although the Ultra 320 parallel still transfers data at a higher rate over a longer distance than the serial versions. However, the serial versions allow for more devices by a factor of six or more.

SCSI's primary use is as a multiple disk interface, particularly in fault-tolerant arrays such as RAID 10 or RAID 5. It is not normally included in modern PC design and must be added through an expansion card.

Name	Bus	Throughput	Max. length	Max. devices
Ultra3	Parallel	160 Mbps	12 m	16
Ultra-320	Parallel	320 Mbps	12 m	16
SSA	Serial	40 Mbps duplex	25 m	96
SSA 40	Serial	80 Mbps duplex	25 m	96
FC-AL 4Gb Fiber Channel Arbitration Loop	Serial	400 Mbps duplex	3 m	127
SAS 3 Gbit/s	Serial	300 Mbps duplex	6 m	16K

Table 3-2. High-Speed Serial Hard Disk Interfaces

Not one of the versions of the SCSI standard ever specified the kind of connector that should be used with a particular interface.

SATA (Serial ATA)
SATA replaces the parallel ATA (now known as PATA) for connecting EIDE drives to PC mother boards. SATA uses a 7-pin serial conductor for signal and a 15-pin connector for power. These are specifically keyed so they cannot be connected incorrectly.
SATA currently has two data rates—.5 Gigabits per second and 3.0 Gigabits per second—and one, 6.0 Gigabit per second, coming into production soon. However, these data rates

have a maximum length of 1 meter and should not be considered for external peripheral connections.

Summary

This chapter considered some of the more commonly used standard serial data communications interface standards. Essentially all data in modern PCs and their peripherals will be transmitted serially. A modem transmits data serially; most networks transmit data serially. There are standards and industry associations for most of these serial architectures.

This chapter has shown how a standard, RS-232 now EIA/TIA 232(F), originally rather limited in scope, has kept pace with technology. Designed well over forty years ago, it is still specified. EIA 442/423 and EIA 485 are standards that describe the electrical characteristics of the transmitters (generators) and receivers. EIA 485 has electrical characteristics similar to EIA 422, except it supports contention—up to thirty-two transceivers—and is used as a network. Lastly, RS-232 was extended to the D suffix, then changed to EIA/TIA 232(E) and then (F) to meet international standards. It will be around for a while as an unbalanced (to ground) standard. EIA 530 is a mechanical standard for a balanced-to-ground signal system that uses a connector identical to RS-232 except in pin-outs.

The evolution of the PC from curiosity to mainstay has brought with it a number of connectivity standards. Though Ethernet (explained in detail in the next chapter) is a serial technology, others, such as USB (now in 2.0), SCSI, and IEEE 1394, will play an important part in the industrial arena as PC technology moves into this environment.

All the standards discussed here are in everyday use, and increasingly in industrial applications where simple, low-cost, reliable communications are required.

Bibliography

Axelson, Janet Louise. Serial Port Complete. Madison, WI: Lakeview Research, 1998.

Electronic Industries Association. EIA RS-232(C). Arlington, VA: EIA, 1968.

—. *EIA RS-232(D).* Arlington, VA: EIA, 1987.

—. *EIA/TIA 232(E).* Arlington, VA: EIA, 1990.

—. *EIA/TIA 232(F).* Arlington, VA: EIA, 2000

—. *EIA 499-A*. Arlington, VA: EIA, 1972.

—. *EIA 422-A.* Arlington, VA: EIA, 1972.

—. *EIA 423-A.* Arlington, VA: EIA, 1972.

—. *EIA 485.* Arlington, VA: EIA, 1982.

—. *EIA 530.* Arlington, VA: EIA, 1986.

Thurwachter, Charles N. *Data and Telecommunications Systems and Applications.* Upper Saddle River, NJ: Prentice Hall, 2000.

4
Local Area Networks (LANs)

How We Got Here

Thus far, we have looked at some communication models and some of the prevalent serial communications standards. In the natural progression from standalone to networked systems we now come to a fork in the road. Originally, all data communications were point to point, much as airplane travel was before the advent of the airport as a hub. There were established networks—telegraph, teletypewriter, telephone, television, railroads, interstate highways—but most data communications was performed over a point-to-point link. Even if those links were made into a network they didn't have a shared medium of transmission.

The need for expensive resources created the need for shared resources. Since bandwidth was (and is) expensive, high-bandwidth lines would have to be shared by many. Printers and mass storage were likewise quite expensive. Facing similar pressure for the use of expensive resources, airlines found that by feeding into a hub and then transporting passengers from hub to hub they could achieve much greater efficiency and reduce shared costs for the passenger and the airline. However, bad things can happen—weather, mechanical problems, and delays by air traffic control—and the efficiency starts to drop off into chaos. In data communications the use of trunks with store and forward switches (the same concept used by railroads and telephone companies) evolved into the wide area data networks.

The aforementioned fork in the road comes at this point: do we want to network locally (as in a plant or office area) or do we need to go through the public network? Do we want to create a local area network or a wide area network? In this unit we discuss the basics of local area networks.

Definitions

For the past 25 years, the process industries have had large numbers of smart devices that communicate electrically to control a process or processes. The combination of such devices and the electrical connectivity that enables them to communicate, by definition, are networks. However, they go under many different trade names, such as data highways, fieldbuses, distributed control system networks (the network puts the distributed in DCS), and so on. Most networks are named after their communications infrastructure or their function. Distributed control systems seem to be everywhere, and every instrument manufacturer has a proprietary variant. In reality, these "islands of automation" are nothing more than local area networks by any definition. An administrative or central data collection com-

puter, or even a process control computer with smart terminals, smart devices, and programmable controllers connected to it, constitutes a local area network.

Programmable controllers, when connected together by a "data highway," comprise a local area network, and when several are connected with a third-party human-machine interface (HMI) to provide integration and a graphics interface, they are a distributed control system (DCS). Any number of manufacturers offer distributed controls with differing degrees of smart peripherals. These too are local area networks. So, what is the definition of a local area network?

There are several. The one this author prefers over most others is: If you own the medium (of transmission), the infrastructure, and three or more devices are in communication, it is a local area network. But another is: a system with more than three nodes that uses a protocol with defined rules and performs a specified set of functions, which are an integrated part of a single business function. It may be suggested that almost any network of more than two devices qualifies. In many ways, that suggestion is almost correct. What then prevents this definition from describing a wide area network such as the telephone system? In most cases, in a wide area network the medium or infrastructure is leased, rented, or otherwise not owned by the company.

A case can be made that a local area network can include wide area network nodes (particularly through remote access services [RAS]). One could say that a local area network is a network performing a similar function (like accounting or personnel). That is one way to view a local area network: a network bounded not by geographic lines, but by functional lines. An example would be a network used in computer-integrated manufacturing (CIM). Regardless of the geographic distances covered by the devices connected into the network, it has one main function: the control of production at that facility.

However a local area network is defined, it will be a network that is not open to public traffic; it is set up and designed to provide a medium of communication for a specific purpose, department, facility, or entity.

LAN Model

Before discussing the LAN model, we should first look at the OSI model and the IEEE 802 model and see how they complement each other. The 802 (named for February 1980, when the committee was first established) standard describes but three layers of function (see figure 4-1).

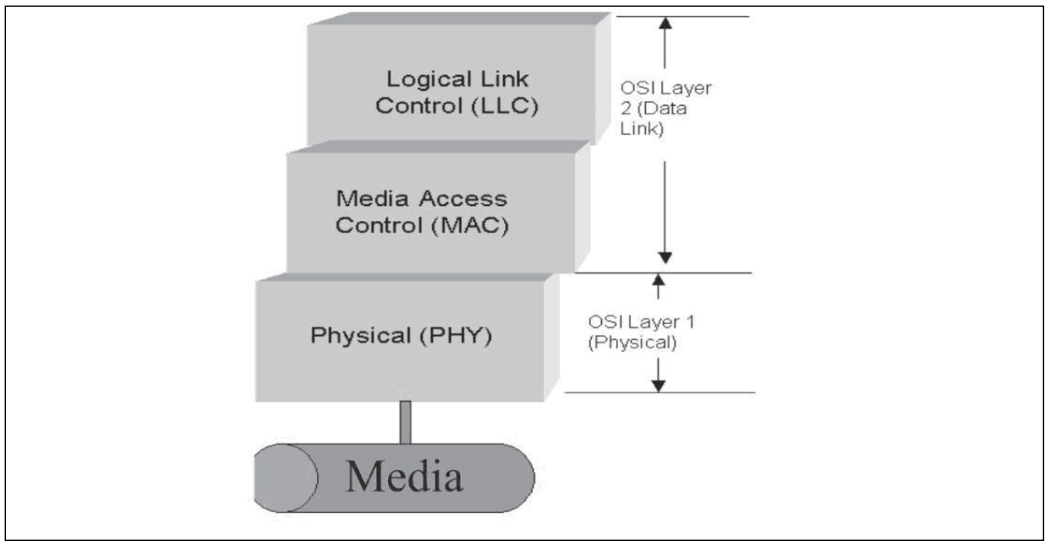


Figure 4-1. The IEEE 802 LAN Model

You will note from figure 4-1 that there are only two OSI layers of function: Layer 2 is subdivided into two layers (called sublayers). These are the Logical Link Control (described in IEEE 802.2) and the Media Access Control (described in most of the other standards, such as 802.3, 802.5, etc.)

IEEE 802.2 (LLC) is about how you interface externally to an 802 system, how you bridge (a Layer 2 function dividing segments of a larger network), and so on. The MAC standards are concerned with who talks and when, how you address each network device on the media, how you frame the data for transmission. The Physical Layer is responsible for placing data on the media, extracting data from the media, and synchronizing the network devices.

Layer 1, the Physical Layer

Placing data on and getting it off of the media:
How data is placed on the media (and taken off) determines the type of network. If data is placed digitally (even though it is conditioned) it will be a baseband system. Typically, if modulation-demodulation are used; i.e., the use of a carrier signal and data modulation (frequency division multiplexing) it will be a broadband or carrierband system.

Broadband

Broadband refers to separating signals by frequency (a process called frequency division multiplexing). The main bus (trunk) connects the devices in the network, and it will have a large number of different signals on it. Broadband services can include actual network technology for multiple local area networks on one trunk, or they can refer to carrier-supplied services such as ISDN and (x)DSL, which are supplied to networks from a wide area network provider. *The bulk of the technology found in broadband local area network systems evolved directly from cable TV systems*, right down to the connectors and 75-ohm cable.

Broadband trunks offer a multitude of services, data, voice, and television—all on the one cable. Using networks on a broadband trunk requires two channels for each network: a transmit and a receive channel. All addresses on the network transmit on the transmit frequency, and all addresses on the network receive on the receive frequency. Obviously, some method for converting the transmit frequency signal is required if the receivers are to detect it. This is the function of the head-end remodulator. Having receive and transmit channels means the bus is directional; that is, all transmit frequencies are inbound (to the head end), and all receive frequencies are outbound (away from the head end). The bandwidth available to broadband is approximately 300 to 400 MHz. This will allow transmission of quite a few network channels as well as other services. Just as you may have two hundred analog television channels on cable TV, you may also have a large number of networks on one cable. Since much of a network's capital outlay is tied up in the cabling and in its life-cycle cost, having a number of LANs on one cable may be a cost-effective solution. Another way to achieve broadband functionality is to use time division multiplexing, i.e., where separate signals are carried in different time slots (used by Telcos in T1 and T3 lines).

Baseband

In baseband transmission, there is but one digital signal on the bus; or shared media. Separation for more than one network is achieved by physical spacing that is using different cables. Most baseband systems are now run on unshielded twisted-pair (UTP) cable, but a few still run on coaxial cable. In the industrial environment they may very well be run on fiberoptic links. Switched media (as opposed to shared) has gained acceptance as a high-performance alternative to statistical-based media access schemes. Switching is accomplished by using a fast-switching matrix that provides a connection to individual nodes, akin to point-to-point technology. Users think they have a private line because the switching is done for only a few packets each switch connection.

Baseband technology has advantages:
(1) no modem is required,
(2) it is easier to install, and
(3) (the overpowering advantage) it is the least expensive per node (at 100/1000 Mbps) by a significant amount.

Compared to broadband, baseband has some disadvantages:
(1) limited capacity,
(2) limited distance per segment.

All disadvantages are outweighed by advantage #3.

Carrier Band

Used in industrial environments, carrier band has only a single carrier on the bus, which is used to translate the digital signal away from a DC reference. Carrier band is a broadband technology, but it has some of the restrictions of baseband, notably, only a single channel.

As a single channel, carrier band eliminates all the costs associated with multiple channel devices, as well as all the capacity. Whether a network is baseband or broadband has little to do with its topology (layout) because either may use a star, point-to-point, ring, bus topology, or combinational topology. Nor does the choice of baseband or broadband have much influence on the access methods used on a network.

Topologies

As stated above, whether a system is baseband, broadband, or carrierband has little effect on how a system is laid out or the system's "topology." In actual terms, most broadband systems will utilize the bus (described in the following paragraphs) while most baseband systems will now be star. That is not to say they are always that way particularly if the medium is fiberoptic where the topology may be some form of point to point (daisy chain or ring) or star.

Topology Definitions

The word topology is derived from topographic of the topographic map. A topographic map shows elevations (the lay of the land) and the topology map for a network illustrates the "lay of the lan." While topology refers to the physical layout, a network system's architecture refers to the ***logical*** connections between both intermediary and end devices. Logical connections may differ significantly from the actual physical connections (the topology). The term *logical* connection usually refers to the way in which the network operating software views the device connections. The three major topologies (either physical or logical) used in local area networks are illustrated in figure 4-2 (star), figure 4-3 (ring), and figure 4-4 (bus).

Star Topology

The star topology has been a fundamental computer network topology since the computer joined networking. Basically (as seen in figure 4-2) there is a central point (or star) and all devices connect only to the star. In one of the first instances of the star topology, a central computer, usually a mainframe or a minicomputer, time-shares all inputs and outputs. Time-sharing is another way of saying "time division multiplexing." The computer's operating time is measured in instruction cycles. In a multiprocessing (or multi-user) system, each task (terminal) is assigned a time slot (a protected location in memory), and each slot has its own program code, data, and the like. The computer uses an algorithm that determines when and how often in the totality of operation its resources are tied to this particular slot.

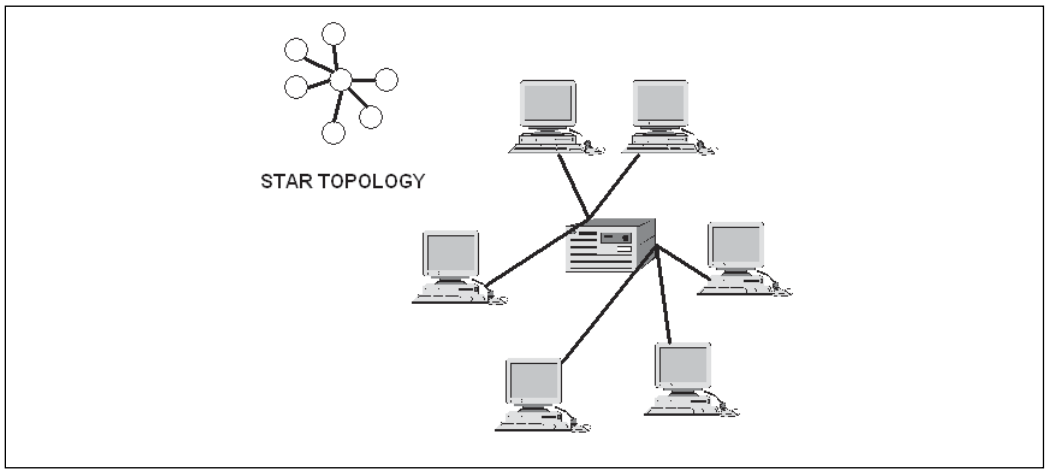


Figure 4-2. Star Topology

In these first instances of star topology all devices input to the computer, and it outputs to all devices. The primary access method was called *polling*. At predetermined intervals, the central point, (the device in control, located at the center of the star in this case the time shared computer) polls each node (terminal) at the end of an arm, allowing each node to transmit and/or receive as necessary during the interval. Note that under the star scheme any node at the end of an arm can only communicate with any other node through the central point.

The star topology is the way computer networks were arranged until the decreasing cost and size of electronics allowed processing power to be spread out among users. The star topology is identical to the topology of "direct digital control" (DDC) systems. These systems were originally hampered by the lower reliability of earlier computer systems. Even today, it is preferable not to be dependent upon a single entity in any system. In loop controllers, regardless of whether there is one or eight loops controlled by the controller or even a PLC, under present practice it ends up being a star topology from the controller or PLC to the field instruments. Certainly, a modern Ethernet network system using a switch, is an example of a star network.

Ring Topology

A physical ring is just what it sounds like: each device in the network is in the path of the data travel. Each device must be active on the ring, or there must be some provision for bypassing the device in the event of failure or inactivity. Token rings (that is, rings in which there is a circulating pattern around the ring) simplify protocol and the maintenance of synchrony throughout the ring. A ring is comprised of point-to-point connections and is ideal for fiber optics.

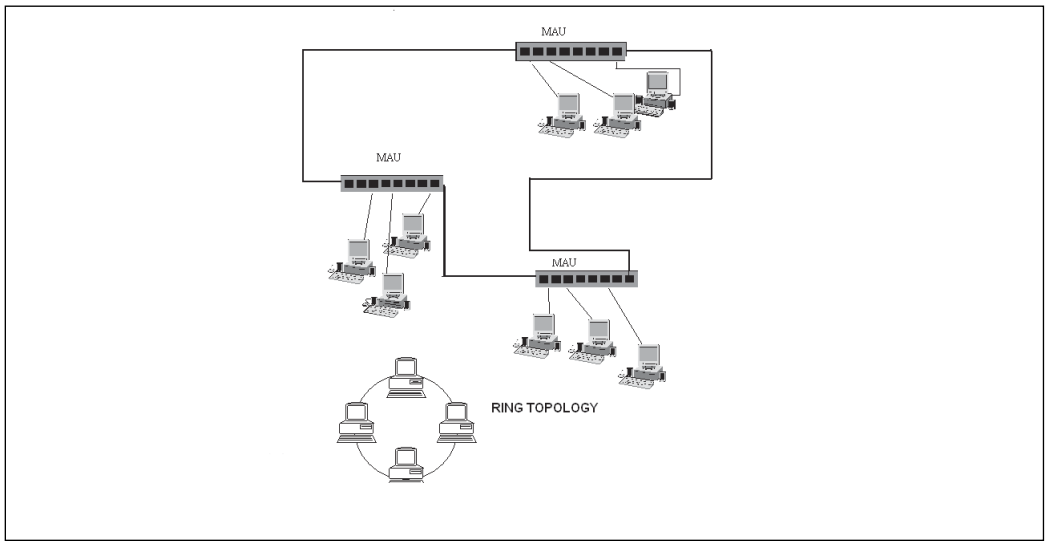


Figure 4-3. Ring Topology

The Media Access Unit (MAU) is the actual device connected in a ring. Each end of the MAU has a Ring In (RI) and a Ring Out (RO) connector. This is the actual physical ring as opposed to the logical connections ring. The computers are connected (logical ring) through electronics that ensure that each computer is on and ready for network traffic before being connected to the ring. If the computer is turned off or otherwise removed, the ring remains unbroken.

Bus Topology

A bus may be a single pair of conductors, or it may look a great deal like the serial extension of a high-speed computer bus. In either case, a bus is a common physical connection to all nodes. The bus architecture is used extensively in industry because of the installation flexibility it affords and because the failure of a node to bypass when it's inactive is not necessarily devastating to network operations. Most industrial bus systems used to be token passers and were used in a logical ring; that is, they were physically connected as a bus. However, the system software has them connected as if in a ring. It should be noted that Ethernet (in the twisted-pair configuration) is a logical bus, but physically it is connected to the hubs/switches as a star.

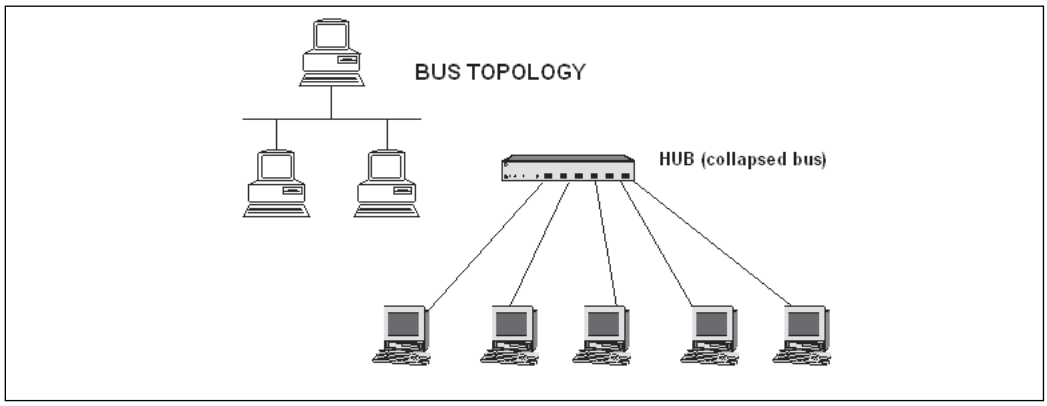


Figure 4-4. Bus Topology

Most modern industrial networks are physically wired in a star (from the wiring hub to the device) but function as a logical ring.

Transmission Media

There are basically three types of media available: copper, fiber-optic, and wireless. In the next section, we will discuss Ethernet media as all three types of media are currently recommended for Ethernet systems.

Ethernet Media

The twisted-wire telephone wiring *was looked at carefully* by early network designers because of the tremendous savings that could be realized if the already installed media could be used instead of special cabling. But given current network speed and interface requirements the media will almost always be a dedicated twisted-pair cable (Category 5 or higher, wireless, or fiber optic) rather then the installed telephone wiring (typically, Category 1-3). As Ethernet assumes a larger share of the industrial interconnect market, its Layer 1 specification, listed in table 4-1 for types 10, 100, 1000, and 10000, will tend to predominate. Table 4-1 illustrates the date accepted, the 802 committee responsible for the specification and the data rate and usage for selected Ethernet speeds.

Date	802.3	Description
1990	802.3i	10BASE-T 10 Megabit over UTP
1993	802.3j	10BASE-F 10 Megabit over fiber-optic cable
1995	802.3u	100BASE-T Fast Ethernet at 100 Megabit
1998	802.3z	1000BASE-X Gigabit Ethernet over fiber-optic cable
1999	802.3ab	1000BASE-T Gigabit Ethernet over twisted pair
2003	802.3ae	10 Gigabit Ethernet over fiber
2005	**802.3-2005**	**Reissue to include previous approved committee amendments**
2006	802.3an	10GBASE-T 10 Gigabit Ethernet over UTP

Table 4-1. Selected 802.3 Amendments

Aside from IEEE 802.Ethernet, shown in Table 4-2, other Layer 1 standards are used, including:

- ISA/ANSI 50.02/IEC 61158-Types 1 and 3/Foundation Fieldbus and Profibus-PA,
- IEC 61158 Type 2, ControlNet

802 and Industrial LANs

An observation about Layers 2 and above: most industrial LANs, until recently, were three-layer LANS, in which the layers were 1, 2, and 7 (the Application layer). They did not need Layer 3 (Network) because they did not intend to go anywhere but on their own network. They were an island of automation. Layer 4 (Transport) was not needed because most networks were connection-oriented and most error handling and packet sequencing was taken care of by Layer 2. All other error handling was taken care of by proprietary actions in Layer 7. Layer 5 (Session) was handled in the proprietary Layer 7 or perhaps by a real-time dispatcher (through Layer 7). Layer 6 (Presentation) was not needed since all machines on these networks used a common syntax.

Having successfully conquered the office and business environment, Ethernet will continue to assume a larger share in the industrial arena. The IEEE 802 standard now encompasses more than local area networks, which are used generally as Layers 1 and 2 in many networking applications. Table 4-2 lists some of the IEEE 802 standards and standard subcommittees (as of 2006). This is by no means a complete list, just a selection of the main committees. Further information can be found on the web pages listed in the bibliography for this chapter. Table 4-2 shows that what started out as a Layer1 – Layer 2 set of network definitions has been extended . Yet 802 is still all about the Layer 1 and Layer 2 implementation and leaves the routing, reliability, and applications interfacing to the higher layers to be defined elsewhere.

IEEE 802.3 is the main concern in this section. We discuss industrial systems including Foundation Fieldbus extensively in the chapter on Industrial Networks. A discussion of IEEE 802.3 is essential here because of the increasing Ethernet applications in the industrial area. IEEE 802.3 specifically describes a baseband signaling system using legacy 50-ohm cabling and has been revised to include 100-ohm unshielded twisted pair (10BaseT, the 100 Mbps Ethernet –100BaseT, the 1000 Mbps Ethernet – 1000BaseT, and 10 Gigabit Ethernet [over fiber]). IEEE 802.4 is not covered in this text (though it was in the first, 1991 edition) since it is obsolete for industrial control (it was the basis of the Manufacturing Automation Protocol [MAP]). It specifically described three types of medium signaling—two broadband and one carrier band—and specified 75-ohm cabling for the trunk cable. (The trunk cable is sometimes called the "backbone cable.")

Standard	Sub-Committee	Focus
802.1		Overview
	802.1d	Specifics for the spanning tree algorithm used in transparent bridging
	802.1p	Traffic class expediting and multicast filtering— allows Layer 2 switches to prioritize traffic at the MAC layer
	802.1q	VLAN
	802.1at	Stream Reservation Protocol (SRP) - QOS
802.2		Defines the Logical Link Control
802.3	-2005	CSMA/CD networks (2005 contains all previous amendments)
	802.3x	Full duplex and flow control
	802.3af	Ethernet-based telephony
802.4		Token-passing bus (legacy systems)
802.5		Token-passing ring
802.6		Metropolitan area networks (MAN) – a dual isosynchronous high-speed ring (inactive)
802.7		Broadband implementations (inactive)
802.8		Fiberoptic network technologies (inactive)
802.9		Standards for integrated voice and data (inactive)
802.10		Interoperable LAN and WAN security (inactive)
802.11		Wireless LAN (Wi-Fi certification)
802.12		Demand priority access (the other 100 MB Ethernet)
802.13		Unused
802.14		Cable broadband implementations (inactive)
802.15		Wireless personal area network (PAN)
	802.15.1	Bluetooth
	802.15.4	IEEE 802.15.4-2003 Low-rate wireless personal area network (WPAN)
802.16		WiMax (Wireless MAN)
	802.16e	(Mobile) broadband wireless access (Mobile WiMax)
802.17		Resilient packet ring
802.18		Radio Regulatory Technical Advisory Group (TAG)
802.19		Coexistence Technical Advisory Group (TAG)
802.20		Mobile broadband wireless access (under study)
802.21		Media-independent handoff (changing seamlessly between different networks, e.g., Bluetooth to GSM to 802.11, etc.)
802.22		Wireless regional area network

Table 4-2. Overview of IEEE 802.(X) Committees and Their Focus

Though IEEE 802.5 has limited availability in the industrial control area, we have covered it here to illustrate the advantages and disadvantages of a ring topology. Most industrial networks are baseband, but not necessarily either 802.3 or 802.5. Most use token passing, although modified Carrier Sense Multiple Access/Collision Detection (CSMA/CD) is also used. CSMA/CD is the method used by 802.3 when using a shared media, but this access method is not unique to 802.3 either.

Wireless LANS

An increasing number of committees are devoted to wireless transmission as it has become a preferred method of installation where installing copper or fiber-optic would be expensive, inconvenient, or impossible. Following are some of the wireless committees and their specifications.

802.11

The original 802.11 standard is now legacy, and there is actually no 802.11 standard anymore due to the marketplace availability of much higher data speeds, just the amendments. The original 802.11 standard, approved in 1997, provided for 1 or 2 Mbps transmission in the 2.4 GHz band using either frequency hopping spread spectrum (FHSS) or direct sequence spread spectrum (DSSS).

802.11b

The first of the 802.11 amendments for which commercial product became available was 802.11b (sometimes called Wi-Fi). It extended the original 802.11 to provide 11 Mbps transmission (with a fallback to 5.5, 2, and 1 Mbps) in the 2.4 GHz band using DSSS. The revised standard 802.11b was accepted in 1999.

802.11a

Standard 802.11a used the 5 GHz band, which provided up to 54 Mbps and used an orthogonal frequency division multiplexing (OFDM) encoding scheme rather than FHSS or DSSS. Though the 5 GHz band is better from an interference standpoint than 2.4 GHz (which is used for portable telephones, Bluetooth, and segments of the amateur band) the higher frequency did not have the same range as 802.11b. This has been greatly improved upon such that time and current adapters have nearly the same range as the 11b devices (though the inability to penetrate structures remains).

Though 802.11a was approved at the same time as 802.11b (in 1999), technological problems and the limited 5 GHz range prevented it from gaining acceptance in the marketplace as quickly.

802.11g

In June 2003, 802.11g was ratified. It works in the 2.4 GHz band (as does 802.11b), but it has a maximum data rate of 54 Mbps (and a net throughput of about 27 Mbps) and is backward compatible with 802.11b. In mixed networks (both 802.11b and g components

together) data speed is significantly reduced (11 and 5.5 Mbps), but alone 802.11g achieves high data rates of 6, 9, 12, 18, 24, 36, 48, and 54 Mbps by using OFDM. Unfortunately, like 802.11b, 802.11g is subject to the same interference in the 2.4 GHz range. The standard 802.11g has been highly successful in the commercial marketplace.

802.11n

At the beginning of 2004 the IEEE formed a new 802.11n task group to develop a new wireless modulation method capable of achieving a data rate of up to 540 Mbps. Proposed standard 802.11n builds on previous 802.11 standards by adding multiple antennas (called MIMO—Multiple Input and Multiple Output) to make possible spatial multiplexing and other coding schemes. Due to competing schemes, 802.11n is still in the approval process and will probably not become a standard until late 2007. However, many suppliers have already produced "pre-n" products for the marketplace using draft standard technology—and the promise to update firmware once the standard is finally approved.

802.16

WiMAX is a "Metropolitan Area Network" (MAN) standard meant to cover much larger areas than the 802.11 standards. Standard IEEE 802.16 was approved in 2002 for operation over a broad frequency range of 10 to 66 GHz. A mobile version, IEEE 802.16e, has also been approved. WiMAX is intended for two distinctly different markets: rural distribution of broadband services and as a contender for broadband delivery of data to and from cellular telephones.

LAN Infrastructure

A LAN consists of devices other than the servers and clients. These generally share a common definition, although sometimes it gets stretched by vendors' or users' usage. We discuss two of these devices—the repeater and the hub—in this section.

Repeater

A repeater is a Layer 1 device that regenerates (repeats) the input signal, thus restoring its amplitude and clock sequence. It is used to extend a particular segment. Care must be taken that the overall delay caused by the repeater propagation and the length of the additional segment do not exceed the network's round trip times. Otherwise the network will time out. The block diagram of a repeater is illustrated in figure 4-5.

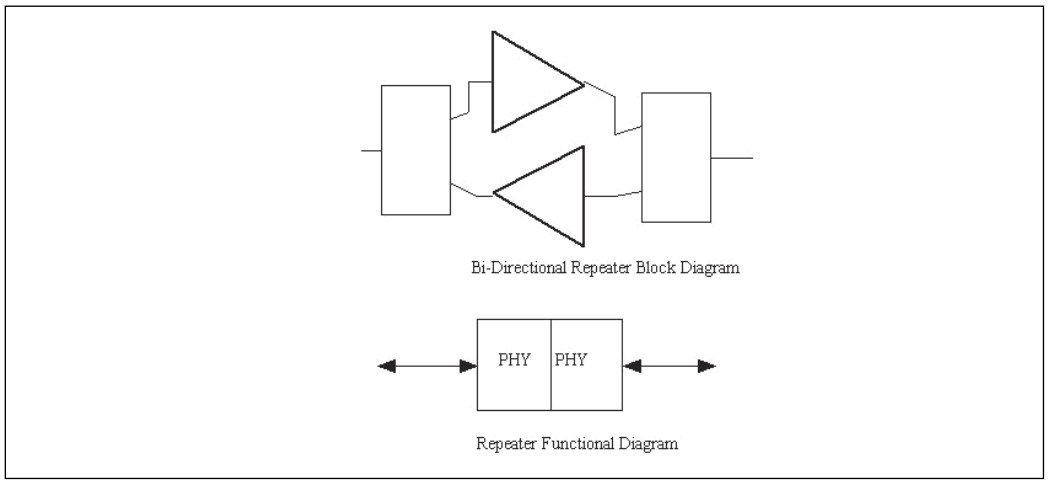


Figure 4-5. LAN Repeater

Hub

Basically obsolete now, hubs first became popular with the advent of 10Base-T Ethernet, which featured a twisted-pair hub consisting of 4 to 24 twisted-pair connectors (RJ-45 in commercial use). Hubs could also take the form of optical fiber hubs, but in either case hubs may be called "wiring concentrators." In the twisted-pair version, what you find behind every RJ-45 connector is a repeater. In simplistic terms, the hub looks like figure 4-6.

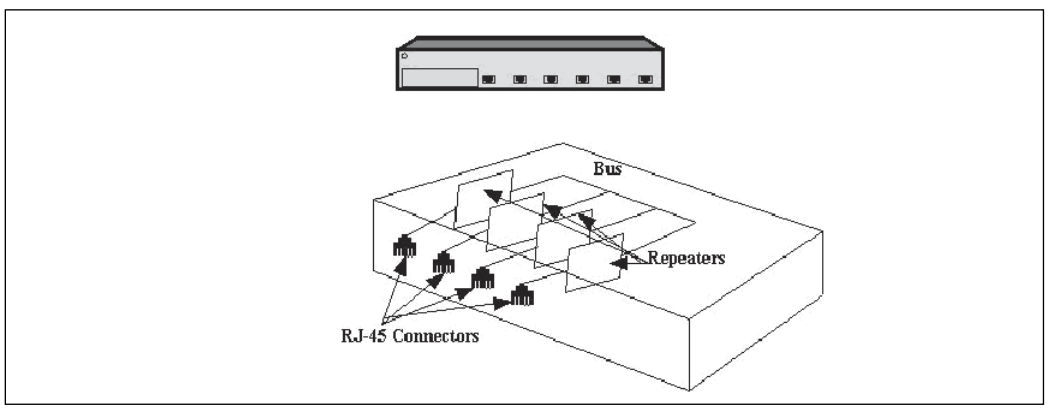


Figure 4-6. Hub: Multiport Repeater

As figure 4-6 shows, each of the RJ-45 connectors is connected to a repeater. It is here that the impedances are matched. Each repeater then connects to the shared media. This is why most Ethernet hubs have power: they are repeating hubs. In fact, in most cases they are called (correctly) multiport repeaters.

Layer 1 and Layer 2 Devices

The limitations of a repeater (multi-port or otherwise) arise from the fact it only knows ones and zeros and attaches no meaning to either state, it just regenerates the signal condition –

outputs a signal with the correct rise and fall time and voltage state achieved. One way to extract better performance is to add intelligence to the intermediate device (between two network nodes) so that the device can read the Layer 2 information, which consists of the destination and source adapter addresses, the "type or length" field, and the CRCC (Frame Check Sequence). The reason is so that the device will know the destination address where the packet is intended to go; the source address so the device knows which adapter sent the packet; the type or length field so the device knows what follows, and lastly, the device can determine if there are any bit errors and either receive the packet or ask for the packets retransmission.

Bridge

Containing two sets of Layers 1 and 2, a bridge is a device that connects two networks of the same type (perhaps to connect different segments or to split a single segment). A bridge provides physical connections, namely, MAC addressing and signaling. Modern bridges are "learning bridges," that is, they look at the Layer 2 addresses and after less than thirty seconds or so they can identify on which side of the bridge a Layer 2 address resides. After that, they only pass traffic across the bridge that requires a destination on the other side. This definitely increases performance by reducing the segment loading. We discuss bridges at some length in chapter 8, on Internetworking. Figure 4-7 illustrates the block diagram of a bridge.

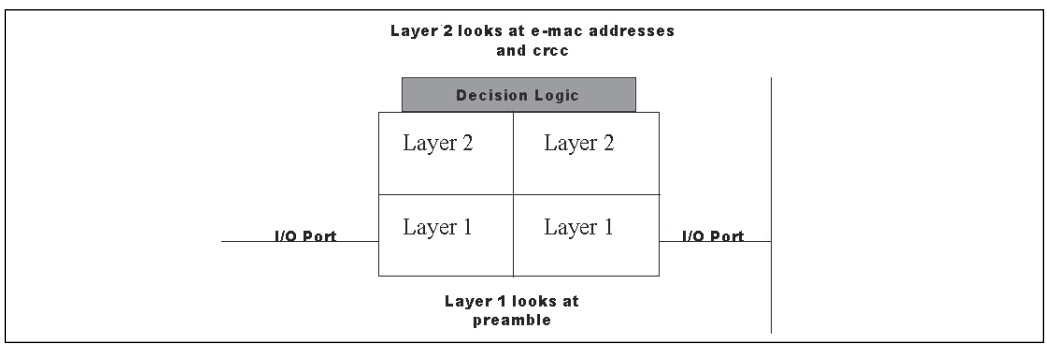


Figure 4-7. Bridge Block Diagram

Ethernet Switch

Found in Ethernet systems, a switch looks like a hub and performs the same function, just in a different way. Instead of having a shared media (bus), the repeaters connect to a switching logic. This logic reads the Layer 2 address and then electrically connects the two ports (sender and receiver) together. So just like a bridge, a switch examines the Layer 2 addresses to decide where the destination node is located. Because logic, memory, and arbitration procedures are built into the switch, each node thinks it has total use of the network. It is no longer a shared media network but a switched one. This has implications that we will spell out in detail through the following chapters particularly chapter 8. Figure 4-8 illustrates a simple block diagram of a switched hub.

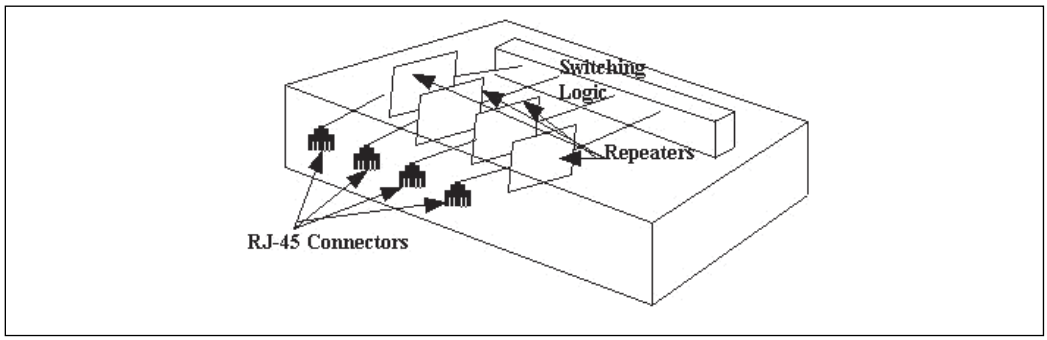


Figure 4-8. Switching Hub

Just like a bridge, the Ethernet switching hub reads Layer 2. Though Ethernet switching hubs and bridges have different functions, in many instances a Layer 2 switch is referred to as a multiport bridge.

Layers 1, 2, and 3 Devices

While reading the Layer 2 information helped us segment our network, it was still only our network with packets being sent to and received from the network adapter (MAC) address. If we should wish to go to a different network, there is no provision to do so in Layer 2. However the type/length control bytes in the Ethernet packet do dictate what follows locating where the network addresses (Layer 3) may be found.

Router

The router contains circuitry that examines Layers 1, 2, and 3 data and will connect different networks, providing network addressing, error correction, physical signal conversion, and conversion to compensate for differences in signal frame size (the size of the data packets). In most networks, the server (file or otherwise) is also a router. In industrial systems the router will be the point of the network that connects to the outside world, be it the communications processor, operator station, gateway, or network monitor point. It will provide routing if the network is capable of routing. We discuss routers at length in chapter 8, on Internetworking. Figure 4-9 illustrates a block diagram.

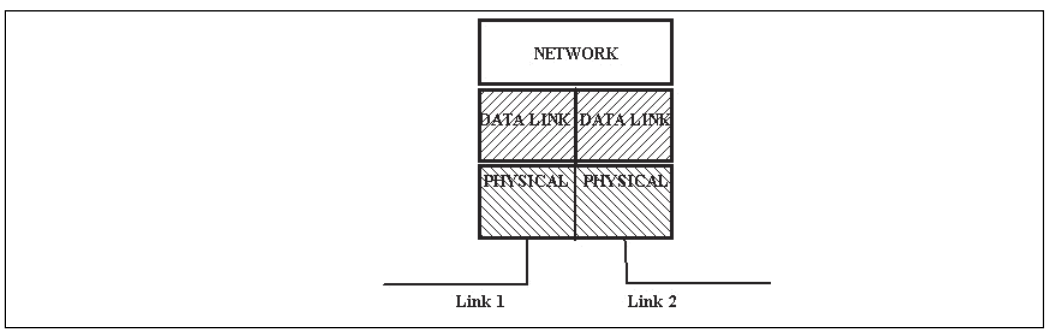


Figure 4-9. LAN Router

Brouter

If we combine a router with a Layer 2 switch we have a departmental router or a Layer 3 switch. This is also called (for marketing reasons, I suppose) a "brouter." This device is capable of reading Layer 2 and Layer 3 addresses. So it knows if you just want to stay on the same physical segment of a network, be bridged to another physical segment (called bridging), or leave for another network (routing). These too are discussed at some length in chapter 8, Internetworking.

All Layer Devices

In the real world systems vary considerably. The organization of ones and zeros on System A may vary considerably with those of System B. The protocols and addressing may be quite different. For this task it takes a dual transceiver, that is, it takes all seven layers for each network to produce the native data which can then be sent back down the other system. Dual seven layer systems are called "gateways." [Except on the Internet a "gateway" is a router].

Gateway

Gateways consist of two complete seven-layer sets. When networks differ widely, as when a proprietary network connects to an open network, a "gateway" is used. Technically, a gateway is a transceiver from each network, with the output of the transceivers under computer control so it may perform protocol conversion, timing, different physical signaling, and the like. Figure 4-10 illustrates the block diagram of a gateway.

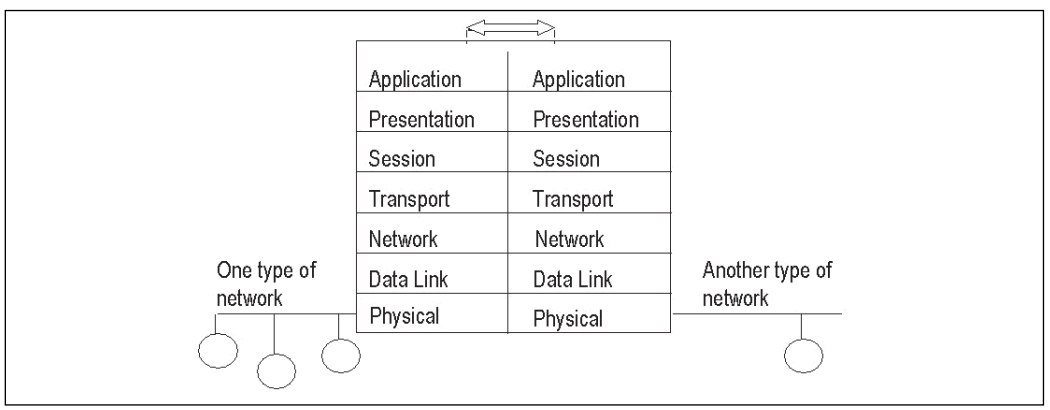


Figure 4-10. Gateway

Layer 2 Functions

We briefly mentioned some of the Layer 2 functions when discussing the bridge. At this point we will take a more detailed look at Layer 2 operation.

Media Access

Access means how a device on a network, such as a LAN, gains control of the network media in order to transmit its information—in other words, who talks and when. Many different methods are available for gaining access, and most depend on network philosophy,

not network topology. Before the rise of distributed processing, multi-drop (and what there was of networks) used the star topology. Peer-to-peer communications were not allowed even if they could have been accomplished technically. It was an era of centralized control. To be fair, the only computing power available lay in a mainframe or a minicomputer. All the terminals were dumb, meaning they had no capability to process data, not that they couldn't transmit. As a result, only one method of access was in general use: polling.

Polling

Though polling may be one of the oldest access methods, it is still used and in certain situations quite effectively. To use a classroom analogy, if the instructor allows no one to speak unless he calls on them, this is polling. In polling, a scheduler, whether a software or hardware entity (it doesn't matter which), determines who should speak and who should listen, and under what conditions. In instrumentation, the *master/slave* access method is an example of polling. Though polling may be used on any topology, it is particularly well suited for the star topology. Polling involves a primary station asking one or more secondary stations, by polling each in turn, if they have any traffic to send. It does this according to an algorithm that is determined by the application. In fact, the master can be rotated among nodes, but the only station that can initiate communications is the master. On most polled networks it is relatively easy to add, delete, or upgrade since only the centralized control or master needs to be updated.

Event-Driven Polling

Polling methods can be modified, an example being interrupt polling, also known as event-driven polling or "hubbing." In this scheme, the centralized control does not query stations but waits until one of the stations announces its intent to transmit by raising an interrupt (or otherwise signaling its intent). It is much like the classroom, where the instructor may (or may not) initiate any communication, but in order for you to communicate you must raise your hand. The instructor then calls upon you. If two of you raise your hand at the same time, the instructor uses an arbitration method (known only to the instructor usually) to select one of you first. In event-driven polling, the central station (or server) is in control and can prioritize simultaneous interrupts. This is generally an asynchronous (i.e., can occur at any time) method and, in general, event-driven polling is how the CPU in a PC is accessed by its peripherals.

Token Passing

In the token-passing access method, a digital pattern (the token) is transferred among its peers. Only the station with the token can initiate communication. Again, it is like the classroom when a piece of paper (the token) is passed among class members. Only when you have the paper can you communicate, and then you must pass it on to the next member. As you can imagine, this will always be a lengthy process, even if you have nothing to say, because you have to have the token passed to you to be given the chance to say nothing. Token passing, however, is "deterministic" in nature. That is, every station knows (within a

window of time) when the token will arrive, so it knows within a specified period of time when it will receive a transmission initiated by another station. Think here of an input to a PID algorithm: it must arrive within a specified time or the algorithm will produce incorrect results. Most industrial networks use token passing because of the deterministic results. Many are further modified so that the station with the token is the master and polls nodes that will never be sent the token, then passes the token on to another master.

Carrier Sense Multiple Access/Collision Detection (CSMA/CD)

Think of the classroom again. Better yet, think of the coffee-break room. When a group of you are gathered together, what are the rules regarding who talks and when? If you have something to say, you wait until the person currently speaking is finished (about a 100-millisecond pause actually) and then you speak. If two of you try and speak at the same time, you have a "collision" and certain rules of order operate (e.g., if it is your boss you generally let him speak before you). Then the opportunity to speak will come (or you may think better of it).

In this system, the node on the network that wishes to communicate listens. When there is an idle pattern on the bus, the node transmits its message. If it receives its own message ungarbled, then it knows (and so does everyone else) that it has the bus. It transmits its allotted times' worth of packets and then goes to idle. Note the simplicity of the scheme: if you have nothing to say, then you are not using bandwidth, are not in the loop, and only those needing to transmit are. Since even at the speeds of electrical transmission a station may start to transmit and another station, unaware due to its electrical distance from this station, may also start to transmit, there will be collisions. When using a duplex switch (all four pairs in the Ethernet cable are used) no collisions will occur, however, since the collision domain is the segment between the switch and the device. The transmit and receive pairs will have no contention. Even half-duplex switched systems only risk the possibility (not very high) of collision if the TX and RX pairs both decide to communicate.

IEEE 802.3/Ethernet: A Layer 1 and 2 Standard

Though IEEE 802.3 is not an industrial protocol as such it is fast approaching critical mass and is used throughout industry. Because of its proven performance, low cost, and high transmission speeds it has become the de facto Layer 1 and 2 standard in many industrial networks, except to the end device (though, in some cases, even there). IEEE 802.3 was not used until rather recently because of myths, facts, and a combination of other factors, primarily the advent of affordable Ethernet switches. As the presumptive heir to the networking throne it bears some detailed examination. It should be pointed out that IEEE 802.3 is a Layer 1 and Layer 2 standard only; Layers 3 through 7 are determined by the particular implementation. Just because two devices have the same Physical and Data Link layers does not mean they can communicate (but it is a start). Most contemporary industrial protocols started life in the commercial area of operations and were adapted to industrial use.

As Layer 1 and Layer 2 standards, the IEEE 802.3 and Ethernet v2.0 have a Physical layer and a Data Link layer. The Data Link layer is broken into two sublayers, the MAC and the LLC (see figure 4-1). The Physical layer is the part of the LAN model that is responsible for making an actual connection and signaling to the medium (trunk cable). In a broadband or carrier-band system, the Physical layer would be the LAN modem (DCE device). In a base-band system, the Physical layer consists of line/signal conditioners found in the network interface. The Physical layer encodes and physically transfers messages between the Physical layers of other units. It provides the procedures and the electrical means for initiating, continuing, and disconnecting actual physical connections.

The Physical layer is responsible for making electrical, wireless, or optical connections to the media, and for actually sending and receiving ones and zeros. In a switched network, the Physical layer is responsible for ensuring the connections. The Data Link layer, and in the IEEE model the MAC sublayer, is responsible for the media access. Though industrial LANs differ greatly in their adherence to any specific model or protocol because of differences between vendors' products, we will concentrate here on the two major LAN media access methods: Carrier Sense, Multiple Access/Collision Detection (CSMA/CD) and token passing. Though this discussion will be on the generic (or standard) configurations, we will cover some industrial specifics (like HART and Fieldbus) in chapter 6, on industrial networks and fieldbuses. Note that 802.3 implements the LAN model: the actual physical connection, the MAC, and the LLC. The following paragraphs include the actual physical description, a brief explanation of the Ethernet modulation scheme or line signaling used, and electrical or optical characteristics.

IEEE 802.3/Ethernet has two older (and now obsolete or "legacy") physical standards for coaxial cable. With the widespread use of twisted-pair cable they are no longer specified for new installation, nor are they recommended in EIA 568, the wiring recommendation for commercial buildings.

Ethernet, since its inception, could run on baseband or broadband networks. The majority of all Ethernet networks (particularly at 100/1000 Mbps) are baseband. In the shorthand used to describe the Layer 3 implementation of Ethernet a specific arrangement is used. First the speed (data rate) then the type (base for baseband, and broad for broadband) then the distance or media type; an example follows.

10Base5—This is interpreted as meaning "10 Mbps, Baseband, 500 meters end to end (per segment)," using RG-223 or RG-8/U. This particular arrangement is referred to as "legacy" meaning it is not commonly used anymore. The coaxial cable used for interconnection is the so-called ThickNet or, if you will, "the frozen orange garden hose," a reference to the difficulty in handling that this particular cable presents. ThickNet/10Base5 requires line taps (usually called "vampire taps"), an active transceiver near the tap, and an auxiliary connection to the network interface card (NIC). The cable impedance is 50 ohms and must be

terminated at each end in a 50-ohm non-inductive precision resistor. Figure 4-11 illustrates 10Base5.

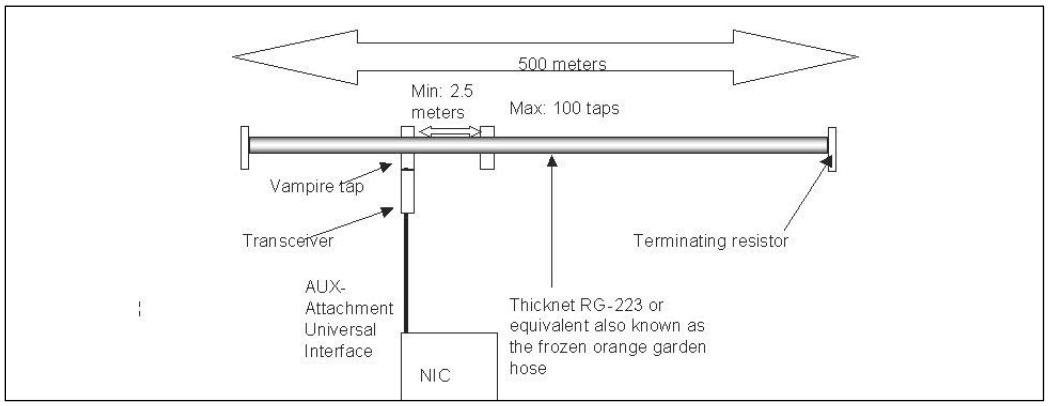


Figure 4-11. 10Base5

10Base2—This is interpreted as "10 Mbps, Baseband, 185 meters end to end (per segment)," using RG58A/U coaxial cable and is also a "legacy" system. This cable is the so-called ThinNet. The coaxial cable is terminated in 50-ohm resistors at each end node, the node being connected by a BNC "T" connector, which is attached to the NIC. ThinNet/10Base2 is usually much easier to work with than ThickNet, but it does have two cables coming into the back of each node, aside from the end ones. Figure 4-12 illustrates 10Base2.

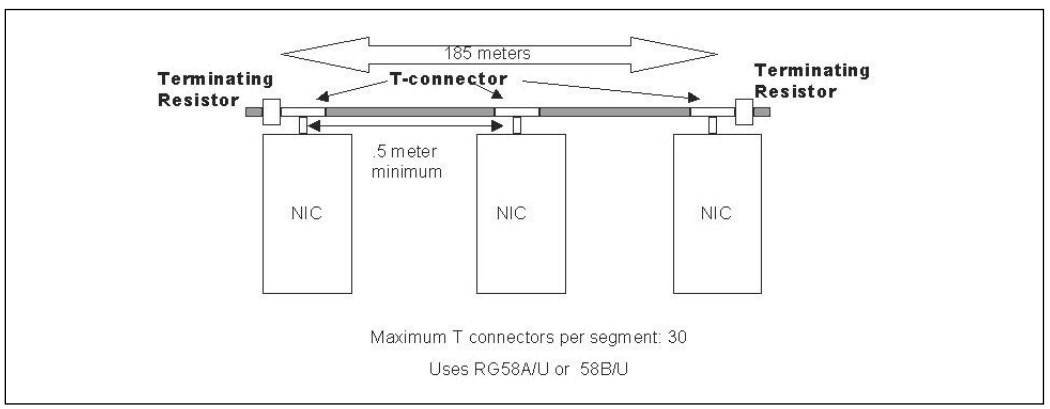


Figure 4-12. 10Base2

10BaseT—This is read as "10 Mbps, Baseband, Twisted Pair," the media being an unshielded twisted pair for interconnection. This system uses a star-wired system whose central point is a hub or switch with RJ45 connectors. The node NIC is connected to the hub or switch with a 100-ohm unshielded twisted-pair (UTP) Category 3 cable, using only two pair of the four-pair cable. The connecting cable may be up to 100 meters in length.

Termination is taken care of in the hub or switch and the NIC, with some vendors calling the connection a segment. Some hubs/switches have eight ports; some have sixteen ports (or more). Active hubs perform management functions, but even passive hubs still require power as they all contain repeaters, as stated earlier. A switch with management functions is called a "managed switch." Figure 4-13 illustrates 10BaseT.

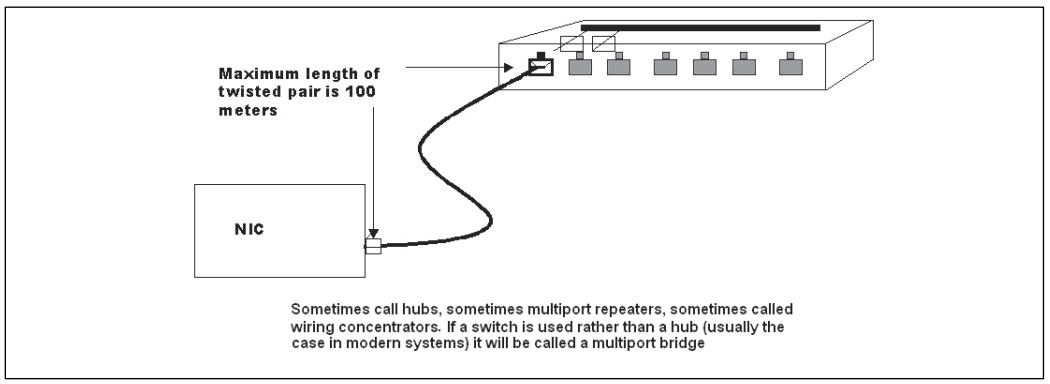


Figure 4-13. 10BaseT

10Base-FL—This is "10 Mbps Ethernet over fiber." It requires one transmit and one receive pair and uses an optical switching hub. Depending on the fiber cable you use, 10Base-FL may have a maximum distance of nearly two kilometers. One frequent application is to connect distant hubs together.

100BaseT—This is interpreted as "100 Mbps, Baseband, Twisted Pair." Generally, the 100BaseT specifications are identical to 10BaseT with the exception that the connecting cable must be Category 5 (or better). Presently, this is the standard Ethernet installation. It requires Category 5 wiring (as opposed to Category 3 for 10 Mbps). Figure 4-12 will represent 100BaseT as well.

100Base-FX—This is "100 Mbps Ethernet over fiber." When run in full-duplex mode 100Base-FX can achieve a distance of two kilometers. Half-duplex is limited to 400 meters.

1000BaseT—This means "1 Gigabit per second Ethernet." Originally specified for Category 5 cable, 1000BaseT requires a tighter specification cable, such as Category 5e, 6, or 6+ cable, to operate properly over copper at 100 meters, its maximum distance. Note that all four pairs of the cable are used.

1000Base-CX—This is a shielded twisted-pair (STP) cable with a 25 (82 feet) meter maximum distance.

1000 Base-SX—This is a short wavelength fiber cable with a maximum distance of up to 550 (1800 feet) meters.

1000 Base LX—This is a long wavelength fiber cable with a maximum distance of up to 5 kilometers (16,400 feet).

10GbE - 10 Gigabit Ethernet A relatively new standard, meant for trunks (high bandwidth inter-network connections) operated originally only over fiber-optics, known as **10GbBASESR, 10GbBASELR, 10GbBASELRM, 10GbBASEER, 10GbBASELX4** (released in 2005). The following are the fiber versions of 10GbE media.

SR (for short range or short reach) was developed to run on multimode fiber for a distance of 26 meters (85 feet). A newly developed 50-μm cable at 850 nm has a 300-meter (984 feet) distance.

LR (long range or long reach) runs over single-mode fiber at 1310 nm, achieves a minimum distance of 10 kilometers (32,800 feet), and typically extends to 25 kilometers (82,020 ft).

LRM uses FDDI 62.5 μm cable for 220 meters (722 feet).

ER (Extended Range) is a single-mode fiber at 1550 nm that achieves a distance of 40 kilometers (131,233 feet).

LX4 (wavelength division multiplex) operates 240 meters to 300 meters (787 feet to 984 feet) multimode to a distance of 10 kilometers (32,800 feet) over single-mode fiber at 1510 nm.

10 Gbps Ethernet over Copper

Due to both monetary and certain installation requirements, a copper version of the 10Gbps Ethernet was developed for unshielded twisted pair. The standard was known as 10GB BASE T and was approved in July 2006 and published in September 2006.

10GB BASE-T—This was originally thought to run over Category 6 (250 MHz) cable, but can actually only achieve a distance of about 55 meters (180 feet). For 10 Gbit Ethernet, Category 6e (500 MHz) cable can be used; however, a new Category 6a cable has been developed (625 MHz - ISO Class E) that will achieve the 100 meter (328 feet) distance of 1000BASE-T. The 6a cable was designed to have more twists, and the twist rate was varied between pairs (to control coupling). This made it a physically bigger cable in part because of the physical separators between the cable pairs. A Category 7 (700 MHz – ISO Class F) cable is recommended for 10 Gbit Ethernet. One version (which uses four pair) the CX4 trades distance for speed and runs 10 Gbits for 15 meters (49 feet).

.

IEEE 802 Medium Access Control (MAC)

The MAC sublayer interfaces the Logical Link Control (LLC) sublayer to the Physical layer. The Physical layer is responsible for the actual sending of ones and zeros. The MAC is responsible for sending and receiving frames. It provides the service of sending and receiving ones and zeros by arranging them into frames. In transmit, the MAC:

(1) initiates transmission (802.3)
(2) assembles the frame
(3) calculates the Frame Check Sequence
(4) sends the frame
(5) ceases transmitting (802.3)

In receive, the MAC:

(1) receives the frame
(2) checks the address
(3) discards other station frames
(4) calculates the Frame Check Sequence
(5) discards bad frames

As you may gather, it is the MAC layer that is primarily responsible for the media access method—token bus or CSMA/CD—hence its name: Media Access Control.

Ethernet CSMA/CD

The Carrier Sense Multiple Access/Collision Detection (CSMA/CD) method is used in many commercial and some industrial LANs, but it defines Ethernet. Historically, there were two "Ethernets": DIX (Digital, Intel, Xerox) Ethernet V2 and IEEE 802.3. However, the minor differences between the two frame assemblies were eliminated in 1997, so there is only one Ethernet now, as defined in 802.3 (2005). Though we will describe Ethernet here on a baseband bus (per IEEE 802.3), it will work as well on broadband buses and wireless systems (from which CSMA/CD was derived). As we have stated throughout this text, one of the main problems encountered on LANs is determining who shall talk and when, and how to keep everyone in synchrony.

Early efforts at creating LANs used a "server," that is, a computerized LAN traffic cop. The server software had algorithms to determine priority, level of access, and so on. It was in reality, and regardless of the way the nodes were connected, essentially a star topology because, as in a star system, when the central computer is inoperable (in this case the server) all LAN operations stop. This could be rectified by using multiple servers or by reducing the server software so the station that was acting as server could also be used as a station and so the responsibility for being server could be switched around. The IEEE 802.3 (and 5) LANs can be server based or serverless. Either may use a net control station, which is easily portable between nodes. The reason for using a net control station is that, while using a "serverless" LAN (peer to peer) appears efficient, somewhere, some single point must have the management responsibility for LAN additions, deletions, security, and so forth. This is not a necessity from a network operating point of view but from a strictly

human and organizational standpoint. In the IEEE 802.3 and 5 versions this management responsibility function is provided by software for any station in the station management partition.

Simplistically, in the CSMA/CD method of control each station listens to the bus when it is idle and discards all packets not addressed to it. When a station is not receiving a message it is ready to transmit. If there is data to transmit at this station it first ensures that there is no traffic in process by listening to the network. Data on the network is sent in relatively short packets so, if the traffic is low, there will be a minimal delay (in microseconds) before an idle condition is detected. If the line is idle, the station will first transmit a preamble (eight octets of repeating 1s and 0s) to allow all stations to synchronize. Then it will transmit a packet. If there is no contention, it may proceed with its own transmission. There's a chance another station may have requested the bus at about the same time, and this station may begin to transmit. This will result in a collision between the signals that will significantly alter each message's content. The first networked station that detects a collision (CD) transmits another set of frames (JAM) to ensure that all stations, regardless of distance on the bus, know that a collision occurred. All stations then drop off the line for a random period of time (in microseconds), which will differ with each station. It is highly unlikely that the same two stations will be in contention when returning from time-out.

Each time a station makes a consecutive attempt to transmit and detects a collision, the drop-off time is extended up to sixteen times, beyond which there is no further extension. At this point, the station determines that either a node has failed and is transmitting all the time (resulting in "jabber"), or there is some other malfunction on the network. Although the CSMA/CD method is used in industrial settings, its use in control applications, and in particular to field instruments such as a transmitter, controller, or valve, is still debated. It is perhaps one of the most efficient methods for lightly loaded networks, that is, networks in which traffic occurs in small bursts, such as office environments. In such an environment, most traffic consists of downloading programs and retrieving or storing documents, which do not impose continuous demand, except, of course, at the beginning and ending of each business day.

Adding or deleting stations (as far as the media is concerned) is simplicity itself. Each adapter will have a unique address, so the adapter just needs to be attached to the network and allowed to communicate. There is no assigned order of polling. Whenever the media access device is told to transmit and the network is idle, it will transmit.

Using the CSMA/CD method in environments (such as process control) that continually transfer data and have a large number of nodes presents some problems. Systems in such environments are heavily loaded, and a large number of collisions could result, particularly if there were conditions that led up to an alarm. In such circumstances, controllers, diagnostics, alarms, and the like would be needed in order to transmit data. A large number of

collisions means a significant amount of time will be spent timing out. Under the CSMA/CD scheme, all message packets would probably get through, but the time it would take for a PID value to be transmitted and received is not predictable. It is possible, although statistically improbable, that the packet might not get through at all. Thus, a CSMA/CD network is called a "nondeterministic" or "probabilistic" network.

The CSMA/CD network's throughput deteriorates under heavily loaded conditions (as do all other forms of network access when loaded). The time-outs in Ethernet are linear, and you can only slow down, not bring down, the network unless you have too many collisions as a result of a jabbering node or an open bus. For these reasons and others, in the past other methods of access have usually been employed in industrial settings. However, one could simply prevent the loading problem by lightly loading the bus through the use of bridges. This would mean that with all nodes requesting permission to transmit, only 10 percent of the capacity would be used. Or you could use a higher layer for real-time control. The current conventional answer is to not use shared media but rather to use a switch. Switched Ethernet is deterministic. The only collision that could occur on half duplex would be if the node and the switch wanted to transmit at the same time, and even that ceases to be a problem if a duplex switch is used (and your available bandwidth is doubled). The only variable now is the latency of the switch itself, which would be a negligible period of time. The loading problem is addressable, solvable, and manageable. This is why you see more and more Ethernets on the plant floor.

Industrial Ethernet is here, ready, and able, and we discus it in more detail in chapter 6, "Industrial Networks and Fieldbuses."

Industrial Token Passing

One of the primary methods of access used in industrial networking is token passing. It is determinate and operates efficiently at heavy loads. However it is a much slower protocol even though it is deterministic.

Token-Passing Bus

The token-passing bus actually does what its name implies. As an analogy, consider a real bus. To get on, you present a token—no token, no ride. The token will be passed to the next rider when the present token holder gets off, so that person may ride the bus. That there will be just one rider on the bus at any one time is easy to see; two tokens aren't allowed. If a token is lost or if the bus waits a little too long at any one stop, a new token is issued. Whoever lost the token may not get to ride the bus without petitioning for a turn to ride. The ride is specified in terms of a number of blocks, and that is all one gets. Trying to ride more than that will result in removal from the bus. So much for the "ride-the-bus" analogy.

Though most of the legacy IEEE 802.4 standard described a "medium" as a broadband unit with repeaters, it should be understood that token-passing buses use other media.

Industrial-use LANs operate baseband or carrier band on such media as coaxial cable, fiberoptic cable, twisted-pair cable, and wireless. The majority of vendors use their proprietary token-passing schemes as baseband, regardless of media. The scheme described next is a generalized concept that a token-passing bus might use (it is the one used in IEEE 802.4).

The token-passing bus is a physical bus topology. However, most token-passing busses are connected logically as a ring (see figure 4-14). One of the important parameters of a token-passing system is the token rotation time—the time it takes the token to make a round trip of the bus. Another important parameter is called "slot time," which is the delay during which station A sends a message to station B, station B processes it, and station B returns a response. This is different from token rotation time. By dividing this slot time by an octet transmission time, the slot time can be determined in octets.

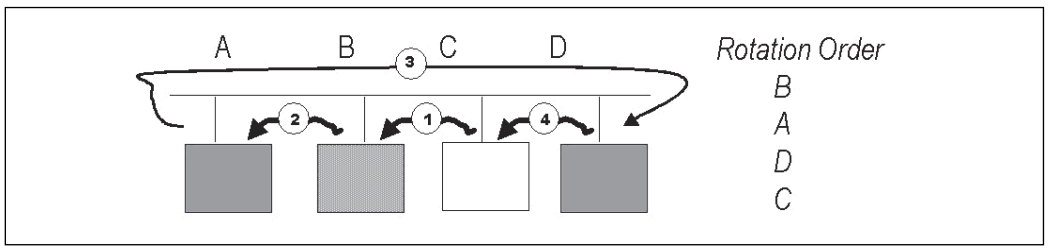


Figure 4-14. Token-Passing Bus as Logical Ring

Before any station can transmit, the following values must have been determined and loaded into the station: (1) the station address, (2) slot time, and (3) high-priority token hold time (each station has a token hold time, and if it expires transmission ceases at the end of the transmitted frame).

Some networks have a four-level priority scheme in effect. That is, starting at the highest and going to the lowest, a station transmits its high-priority packets first. If there is time left on the token-hold timer that corresponds to that priority, then the station can transmit its lower-priority packets. Without the priority scheme, all messages have the same high priority, and all use the high-priority token timer. New stations can be added using the response windows methods. At intervals, a "solicit successor" frame is sent and allows one response frame. In the solicit successor frame will be the range of addresses for stations between the issuing station and the next lower station, its successor. Stations whose address falls into this range can then respond with a set successor frame. If the issuing station receives a valid set successor frame, it places that address as its successor and passes the token to the new successor. If more than one station responds, contention will result. It will be settled through a procedure that ensures that only the appropriate successor is chosen using the resolve contention frame. The order of token passing goes from the highest address to the lowest. Therefore, to find a successor, the station that issued the solicit successor has to look only at addresses lower than its own. In the end, the lowest of

the lower addresses will be selected, and the process will repeat itself. However, this process adversely affects token rotation time because the token has been retained by the station that issued the solicit successor command. A solicit successor second frame is used when the recent lowest address station seeks a new successor.

There is also a "ring maintenance" rotation time. If the token rotation time is slower than this time, no additional solicit successor messages will be transmitted until the token rotation time is less than the ring maintenance time.

Token passing, then, is a relatively easy-to-implement protocol. It is at its best when it's heavily loaded, because each station is assured of a chance at the token. If token passing is lightly loaded, it is inefficient because it gives every station a chance, even though they may have no information and will merely pass the token along. It would be more efficient to service only those stations that have traffic (as in CSMA/CD). Token passing is deterministic (round-trip time is predictable) under heavy loads, which is precisely the reason why it is used in industrial environments.

802.5 Token-Passing Ring

The token ring uses token-passing protocols that are similar to the token-passing bus, and it provides quicker response and better priority handling but at a higher per-node cost (very much higher compared to 802.3). A ring topology suitable for industrial use should meet the following criteria:

(1) Have a 16 Mbps data rate or higher,
(2) No single failure can bring down the whole network,
(3) Failure of a cable or transmitter should not disconnect the station (i.e., a redundant ring),
(4) Offer support for more than thirty-two stations,
(5) Provide reasonable per-node costs.

At the time of writing, the token-passing ring is capable of 100 Mbps. Because 802.3 was widely available at 100 Mbps for less cost than a 16-Mbps ring, the ring standard for 100 Mbps was finally approved in 2000. Due to media expense and per-node cost, this topology has yet to find its way into many industrial settings. The interesting thing about a ring is that it is receiving the same frame as it is transmitting (even at half the speed of light and with a 30-mile radius, not much time elapses in the round trip). This allows the transmitter when it has the token to set the "I have data" bit and attach data. As it is processing the transmit frame, it may read the location where the destination station accepted the frame and change the bit on the received frame to a token, remove the data, and pass the token on to the next station.

Logical Link Control

Together, the MAC and the Logical Link Control (LLC) are responsible for placing and retrieving information without errors on the lower, Physical layer. The specifications in IEEE 802.2 outline the Logical Link Control's responsibilities, regardless of which media

access/topology is employed, for any 802.X-compliant network. Most proprietary networks have their own version of Layer 2. However, for the purposes of discussion, the 802.2 LLC will be examined here.

The functional definition of the Data Link layer includes framing data blocks or packets (measured in 8-bit octets), determining the check character, and determining network addresses. The two sublayers, MAC and LLC, must ensure error-free (at the bit level) data transmission and reception. The MAC receives instructions from the LLC and performs protocol functions along with frame (packet) error checking. The Logical Link Control section interfaces the user program and provides the MAC's services to the user. It interprets the data frame for Link Service Access Points (LSAPS), these being the source and destination LLC (usually); it also identifies the type of service.

Type of Service

A connection-oriented service means a service that is point-to-point oriented. More precisely, a connection-oriented application starts out by acquiring a data channel. Once connection is established, data transfer is effected. This may be a node-to-node, end-user-to-end-user, or one-layer-to-another-layer transfer. On the other hand, connectionless service places addressed packets on the media (Datagrams), without establishing a data channel (called send and forget). Both 802.3 and the Internet are essentially connectionless. The rest of this section discusses the three types of service.

Type 1: Connectionless service allows two LLCs to exchange data without establishing a data link. There is no message sequencing or error recovery, both of which are taken care of by the higher layers. This essentially hands off end-to-end reliability (not the CRCC, which is a node-to-node error correction) to a higher layer, the Transport (Layer 4) or Application (Layer 7) layer.

Type 2: Connection-oriented service establishes a data link and provides message sequencing, flow control, and error recovery. It does not require the services of a higher layer, performing all these services in the Data Link layer. In other words, there will be no Datagrams with Type 2. Industrial systems, because they need acknowledgment and have less overhead, have typically used Type 2 even on token-passing buses.

Type 3: Connectionless, Acknowledged Service is limited to frame acknowledgment, limited flow control, and recovery on a single-frame basis and will be limited to a single (not routed) segment. This form is the one adopted for most modern industrial systems. The single-frame basis means that the transmit station will keep its last transmitted frame until the addressed station acknowledges this frame. The transmit station will retransmit this frame a set number of times and then assume that something is awry. Automatic or manual recovery methods (such as raising an alarm so as to bring a human) will be needed to determine the reason for the lack of acknowledgment. The software could just as easily

determine that, due to the lack of response, the called station was out of service and so apply maintenance procedures.

Classes of Service

Class 1 service supports Type 1 operation only; Class 2 service supports Type 1 and Type 2; Class 3 service supports Type 1 and Type 3. Class 3 is the one preferred for most token-passing buses in industrial applications.

802.2 Information

Whether you are using 802.3 or 802.5 (or any other 802.X), the 802.2 information is identical at the LLC interface. This information comes after the Physical layer information (after the length or type octets). Figure 4-15 breaks out the 802.3 frame. The 802.2 data is located right after the type/length octets in the block labeled "Data." Though the two type/length octets could indicate that TCP/IP data follows, if the decimal value of the 16 bits is 1500 or less (to 64) octets it indicates the length of the frame, and that 802.2 data follows.

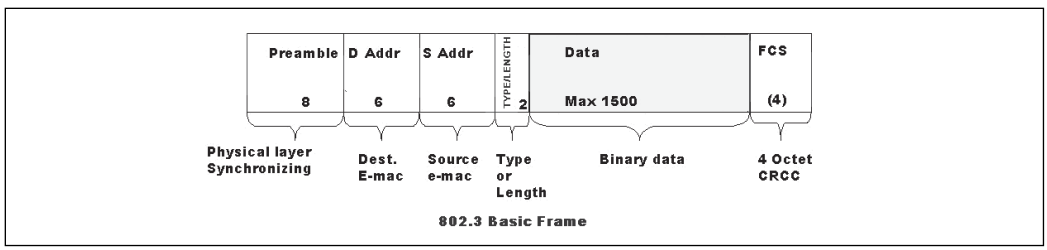


Figure 4-15. 802.3 Ethernet Frame

In concept, IEEE 802.5, token ring, has a similar frame (although the number of octets and the use of a poll octet instead of a type/length are both different), and by the time the information reaches the LLC it is the same interface. That is the rationale for 802. In the 16-Mbps version of 802.5, the frame length can be up to 17.1K bits (or about twelve times the 802.3 frame size), although there is a move afoot to make "jumbo" frames for 802.3 of 9 Kbits. At the LAN level, the 1,500-octet frame size is not necessarily constraining given modern processor speeds and the off-loading of network functions to the network adapter. However, it does use up processing time on overhead. Many Gigabit and 10 Gigabit switches (at the high end) will support frames in excess of 9,000 (some as high as 14,200 octets). Since the frame size has the most deleterious affect on WAN performance it appears to be but a matter of time before the jumbo sizes (for higher-speed Ethernet) are specified.

It is the type/length fields that are the most interesting of the Ethernet frame. If the value of these two octets is less than 1,500 decimal this indicates that an 802.2 LLC frame (usually three or four octets) will follow. An Ethernet frame may be 1,518 octets (1,522 if 802.3q-VLANs is used)). Some exceptions to this rule were made for compatibility with then-existing systems. For example, one pattern after the LLC is for SNAP, and another is for Novell (two octets of all ones—the Netware CRCC). Neither will be discussed here (a great

deal of reference material can be found on the Internet by pointing your search engine to "SNAP" or "IEEE 802").

If the decimal value of the two octets' contents is greater than 1,500 (or the HEX value is 05DC), the protocol can't be the length, so it must be some other protocol that follows (other then 802.2). Of particular interest to us is if the decimal value is 2,048 (Hex value 0800) as this indicates that the Internet Protocol will follow (a minimum of twenty octets in IPV4). There are many others available protocols (XNS IPX, etc.) but most have yielded to the de facto standard of TCP/IP.

LAN Layer 3 and 4 Software: TCP/IP

First, it should be acknowledged that up until recently, most industrial systems didn't need a Layer 3 (Networking). They were never intended to leave their "island of automation." With the advent of modern customer-driven, partnered, just-in-time manufacturing the need for plant-wide (the word used now is *enterprise*) communications became evident. Some industrial systems make it possible to move between several industrial networks, but that is more akin to bridging than to TCP/IP's intent. And though we discuss routing in detail in chapter 8, it should be stated here that a routed system is much more reliable and is much easier to converge (recover after a link or device fails).

Transport Control Protocol (Layer 4) and Internet Protocol (Layer 3) are two of a suite of protocols developed for the Department of Defense's Advanced Research Projects Agency project. They were intended to be the lowest common denominator for connecting diverse systems together in a private wide area network of considerable speed (for that day and age, the 1970s). These protocols were designed with the then systems in mind and certainly not for what the Internet has evolved into. It is a myth that the system was designed from the start to withstand a nuclear attack. At the time it was being introduced as a routed system. A paper not connected with ARPA was the one that touted the fact that routed systems were best able to withstand a nuclear attack and its infrastructure damage. A routed system is segmented and routed so alternate paths could be established if the primary path had become disabled. ARPAnet is now the Internet, but, remember, it has design roots in equipment from several technology generations ago. It was never intended to host as many nodes as it now does and certainly was never (originally) intended to run on a local area network.

There are many advantages to TCP/IP. It is an open, free, and standard. It is user driven. Almost all major operating system vendors had and have the protocol stack included in the system at no extra cost. It is the language of the Internet. It provides for a robust method of reliable data transmission and reception. However, it has its disadvantages as well. When running on a local area network it requires considerable overhead, has known security holes, and is not a real-time system because it uses a store-and-forward method.

How did this become the standard? By default. It was here, worked, and was and is understood. The only competitor (other than proprietary systems, whose vendors either charged for their protocols or designed them so they could only communicate with that vendor's machines) were the OSI-compliant systems. However, between the time when the standard specifications for the OSI-compliant systems were completed and the time when cost-effective hardware became available, the window of opportunity was slammed shut, and only one viable technology existed in hardware, software, and specification: TCP/IP.

Remind yourself that Ethernet (IEEE 802.3) is a Layer 1 and Layer 2 set of protocols. TCP and IP are Layer 3 and 4 protocols. Most industrial networks did not use Layer 3 or 4 but went straight to Layer 7 (Application). IP is currently used widely as IP v4 (version 4). IP v6 (version 6) is slowly being installed. To understand why version 6 is needed, we must first look at version 4.

Figure 4-16 is a diagram of the IP v4 header. Here we are primarily concerned with the network addresses, the thirteenth through twentieth octets. Notice that they are thirty-two bits in length.

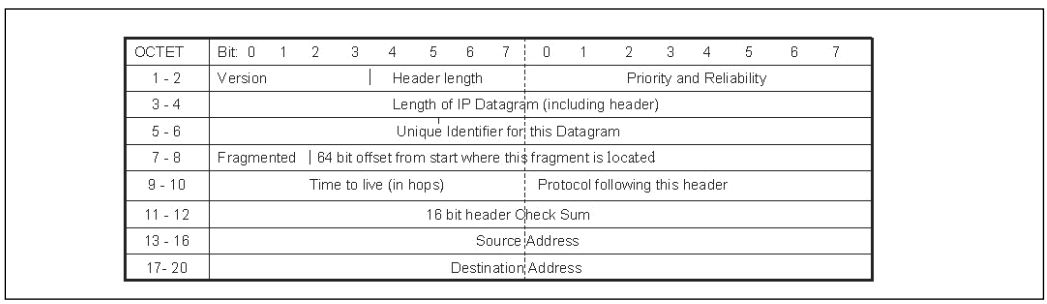


Figure 4-16. IP v4 Header

The addressing scheme was originally intended to help routers perform. Routers originally had tables that were hand-entered, and their processing speed was not as fast as it is today. If some way could be devised so the router could determine the processing needed in the first of these four octets it would benefit the performance category.

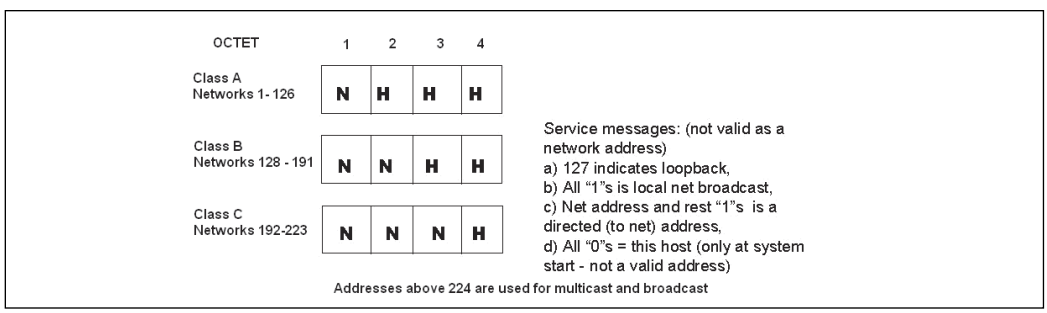


Figure 4-17. Illustrates the General Addressing Scheme IP v4.

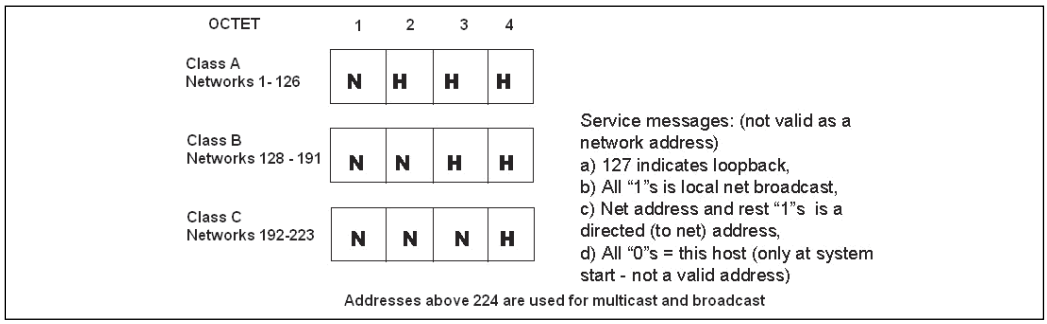


Figure 4-17. Class Address Scheme

Figure 4-18 illustrates how the addressing scheme divides up networks and hosts (nodes).

Class	First Network Address	Last Network Address	Max Number of Networks	Max Number of Hosts per Network
A	0.X.X.X	126.X.X.X	126	16,777,214
B	128.1.X.X	191.254.X.X	16,384	65,534
C	192.1.X.X	223.255.254.X	2, 097, 152	254
D	224.X.X.X	255.255.255.255	N/A	N/A

Figure 4-18. Addressing Possibilities

Though the IP v4 addressing scheme seems adequate for any industrial plant—after all, a couple of Class B addresses would be all that is needed—the designers did not foresee the day when every computer, smart device, telephone, and probably microwave and refrigerator, would require an IP address. The class A addresses have been returned, and a classless method of addressing is in effect, yet even then there will not be enough addresses for the anticipated usage (as in the whole world). Enter IP v6. This version corrects and updates IP v4. First, it contains sixteen octets (128 bits) each for the source and the destination address. That should keep a number of free addresses available for a while. Figure 4-19 illustrates the forty-octet IP v6 header.

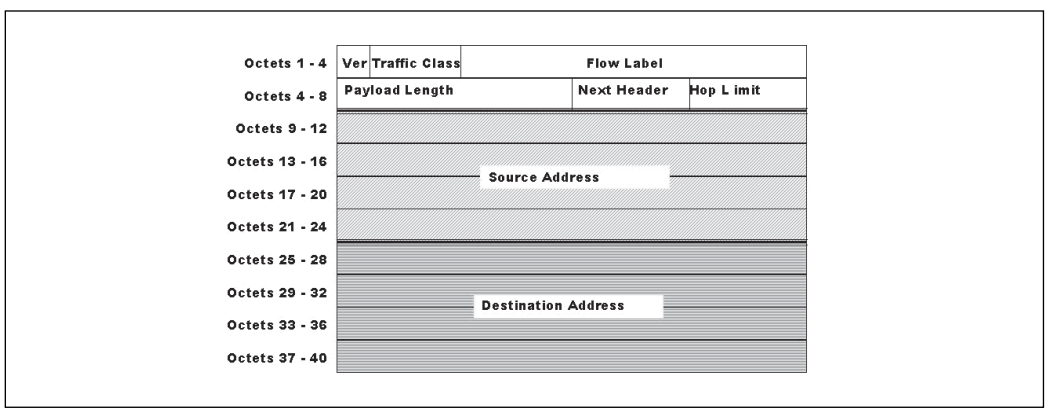


Figure 4-19. IPV6 Header

The various bits are assigned meaning. These meanings are discussed next.

Version (4 bits) will obviously be a 6 (binary value) for IP v6 and a 4 (binary value) for IP v4. During the transition from IP v4 to IP v6—which may take a decade and a half—devices will need to know what header version they are to process. That is why the version octet is the first in both types of IP headers. The minimum length of the IP v4 header was twenty octets. The only length of the IP v6 header is forty octets.

Traffic Class (4 bits) is a priority system, which states whether the packet can suffer delays or is to be treated as streaming (real-time) traffic. Values from 0 to 7 are capable of delay, while 8 through 15 are for real-time traffic.

Flow Label (24 bits) is to be used for virtual private networks (VPN) or other pseudo connections that have specific requirements.

Payload Length (16 bits) indicates the number of bytes in the packet following the header. The header is not included in the length count.

Next Header (16 bits) indicates which type of extension header (this makes the IP v6 header extensible), if any, follows this header. If no header follows, then Next Header indicates what protocol (like TCP) is following the header.

Hop Limit (16 bits) serves the same purpose as the Time-to-Live field in IP v4 does in practice. Though TTL could have been measured in seconds, it almost always was measured in hops (router connection). In IP v6 the practice is merely substantiated, and it states the number of hops the packet can be relayed.

Source Address (128 bits) is the IP v6 address of the sending device.

Destination Address (128 bits) is the IP v6 address of the receiving device.

Another Layer 3 Scheme: OSI-IP

The OSI Layer 3 is considerably more detailed than the protocols described thus far. The U.S. government made a considerable push to adopt the OSI protocols (GOSIP). However, mandating a policy rather than allowing the marketplace to determine what will be standard is fraught with peril. One might very well recall the competition between the Betamax and VHS videocassette technologies. A number of sources thought Betamax very much technically superior to VHS. Price, number of suppliers, and superior marketing made VHS the overwhelming favorite, technical superiority or not. OSI routing protocols are very well thought out and certainly much more robust than either IP(v4) or IPX. They are an international standard, and an open one. But they did not prevail in the TCP/IP world, superior, robust, or not.

OSI IP has three components: End Systems (ES), Intermediate Systems (IS), and the Routing Domain (RD). An End System is the OSI term for the end user, which is some form of computing device. An Intermediate System is typically a router. Addressing for the OSI system is complex, as OSI is designed for just about any communications topology (LAN, WAN, and everything in between). It is hierarchical and uses variable-length addresses ranging from 16 to 160 bits (20 octets).

LAN Layer 4 – Overview

Layer 4 is concerned with end users only (a gateway with seven layers is also considered an end user). Intermediate nodes (such as bridges and routers) have no use for Layer 4 or any higher layers. Layer 4 is necessary when the Layer 2 is operated in a connectionless manner. In most industrial networks of the three-layer variety protocols are either connection oriented or the application layer handles the packet sequencing and accountability. With the encroachment of public standards upon the proprietary industrial networks, the standard used most is the Transport Control Protocol TCP of the TCP/IP combination. It is the one we explain next.

TCP/IP

As we've seen, TCP/IP, an acronym for Transmission Control Protocol/Internet Protocol, was developed by the Department of Defense's Advanced Research Projects Agency to work across dissimilar computer platforms, ranging from micros to mainframes. As routing of LAN data became desirable, proprietary schemes were used to fill the functions of Layers 3 through 7. In order to use an open set of protocols, LAN vendors began using IP. IP interfaced to higher layers using TCP, with the lower layers (notably IEEE 802.3) and to peer, gateway-to-gateway transactions.

IP was designed to be as flexible as possible. It therefore offers very little in specific services using the unreliable Datagram service. In other words, it only supports connectionless type service, so no error checking (other than frame bit checking by the CRCC) is provided. TCP, a Transport Layer 4 protocol, did the packet sequencing and counting along with other higher layer functions. TCP/IP has found wide use and is widely adapted to LANs, particularly those with mixed operating system platforms. TCP/IP is built into most versions of UNIX, Linux, Windows 9X and 2003/XP/Vista, Macintosh, Solaris, and other operating systems.

On LANs, TCP/IP finds most use in Ethernet (IEEE 802.3)-type networks, although it is usable on any network. The primary difference between OSI-compliant standards and TCP/IP types is that the OSI model encompasses and attempts to standardize all possible elements so a manufacturer knows precisely what to put into a data communications product. TCP/IP, on the other hand, starts out with very few rules as standards and allows add-ons whenever necessary. Though some add-ons are not universal, and therefore not everybody can use them, perhaps not every station needs to use (or is hardware capable of using) the add-on. The basic TCP/IP still works.

The Transport Control Protocol (TCP) program assumes that the lower layers only offer unreliable Datagram service. This means that the TCP program must provide the transport services of error recovery, packet sequencing and flow control.

There are really two different transport services: UDP (User Datagram Protocol) and TCP. TCP establishes, maintains, and terminates connections between processes. TCP provides an acknowledgment process for reliable packet delivery and performs sequencing (packet assembly and disassembly) using a duplex mechanism. UDP allows a higher layer to concern itself with the reliability; it is a send-and-forget packet.

The IP itself takes care of the Network layer, allowing for packet fragmentation and reassembly. A companion program, Internet Control Message Protocol (ICMP), which is actually an extension of the IP, reports unrecoverable errors to the TCP program. Figure 4-20 illustrates the TCP header, and Figure 4-21 lists the TCP protocol elements.

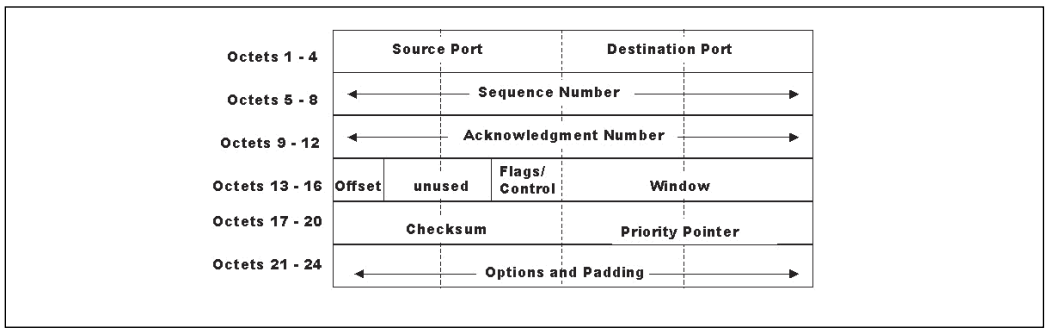


Figure 4-20. Location of TCP Information

Source Port	Originating port
Destination Port	Receiving port
Sequence Number	Byte Counter integer(counts bytes in sequences)
ACK Number	Acknowledges previously received packet sequences
Data Offset	Number of 32 bit bytes in Header
Control Bits	Determines TCP packet type
Window	Size of data this segment will accept
Priority Pointer	Points to byte succeeding an urgent pointer (set by control bit)
Options	Determines end of list, no-op,or max segment size.
Padding	Fills out segment to next 32 bit boundary

Figure 4-21. TCP Element Definition

UDP, the other Layer 4 option with TCP/IP, is used quite often because it is a connectionless transfer (TCP itself can be considered connection-oriented) and has very low overhead. UDP is used in industrial applications where the higher layers perform the reliability checking, and correcting. A UDP frame is displayed in figure 4-22 for comparison with the TCP frame in figure 4-21.

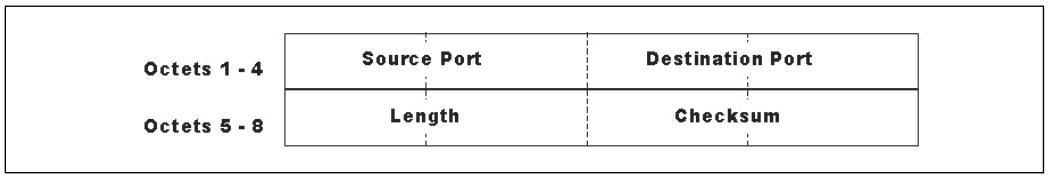


Figure 4-22. UDP Frame

Of course, TCP isn't the only Layer 4 protocol. There is the OSI TP and Novell's SPX. The OSI Transport layer provides five classes of transport, four of which require connection-mode network service (CONS). The vast majority of OSI installations use connectionless transport (CLNS) and do not (of course) require CONS service. A list of the bit sequences is not provided due to the complexity and options they involve. Two features of OSI are worth commenting on: congestion avoidance and flow control window adjustment. The congestion avoidance algorithm allows the Network layer to notify the Transport layer when congestion is detected, for which the Transport layer can reduce its outstanding packet balance.

If the Transport layer determines that the network has lost a packet, it can reduce the outstanding packet window to one, then increase the balance by one each time an acknowledgment is received. This process is called flow control window adjustment.

LAN Layer 5 – Session

The upper three layers, starting with Layer 5, are usually considered data processing more than data communications. Typically, these functions are found in the network operating system. Integrated and not well structured into finite layers, most network operating systems perform these communications functions as part of their core services and usually through a protocol stack.

LAN Layer 6 – Presentation

Again, this function is usually integrated into the operating system but may be accessible for encryption (Crypto API) or for other forms of syntax conversion.

LAN Layer 7 – Application

LAN Layer 7 is where the data is interfaced to the application software. There are many utilities (file systems, directory services, e-mail, etc.) or services running that support various applications. These utilities or services are very much part of the operating system or proprietary Layer 7 for industrial LANs. Ethernet can be the Layer 1 and 2 standard, and TCP/IP can be the Layer 3 and 4 standard. There is a great deal of confusion about how to standardize Layer 7 services. On one side of the commercial LAN world, you have the UNIX camp, which uses the Network File System (NFS), and in the Windows camp you have the Server Messaging Block (SMB). The industrial Layer 7 arena is no better, with confusion and no particular standard prevailing. Even if your systems were standardized from 1 to 4, it doesn't mean they will talk to each other if they have different Layer 7 protocols.

The IEC 61158 standard for fieldbuses defines Application layer standards for each of the service types. Of these types, the most common are Foundation Fieldbus (Type 1) and Profibus (Type 3), which share a common Application layer called FMS. Another commonly used protocol is ControlNet (Type 3) and its Ethernet equivalent, EtherNet/IP, both of which share CIP (Control and Information Protocol) at the Application layer. Modbus/TCP also has a well-defined set of Modbus commands at the Application layer.

Summary

In this chapter, local area networks were discussed in concept from the vantage point of Layers 1 through 4. The majority of industrial networks use just Layers 1 (Physical), 2 (Data Link), and 7 (Application). However, with the growing emphasis on communicating across the entire "enterprise," routing and transport layers are becoming more widely utilized. We also discussed Routing and Transport protocols, as well as the different access methods. Structures of various LAN protocol frames from Layer 1 through Layer 4 were also examined. The reader should be aware that beyond Layer 1 you need detailed knowledge of a particular protocol to achieve more than a conceptual understanding. Additionally, in most cases, these protocols are not adjustable, usually even to system programmers, so monitoring and analyzing are the usual procedures for determining irregularities or discontinuities in service.

In addition to 802.3, this chapter also discussed a number of other protocols, including the newer wireless ones under the 802 umbrella. These will be of increasing importance to industrial settings insofar as wireless technologies offer some unique advantages in application.

The wealth of information you can gain by monitoring these protocols is reason enough to seek deeper understanding. If you want to study the many topics in this chapter in greater depth, an Internet search engine will provide you with all the data and perspectives you need.

Bibliography

Note that Internet links may change.

Deering, S., and R. Hinden. *Request for Comments: 2460 Internet Protocol Version 6 (IPV6) Specification.* December 1998.

International Electrotechnical Commission (association website). http://www.iec.ch.

Institute of Electrical and Electronic Engineers (association website). http://www.ieee.com.

Institute of Electrical and Electronic Engineers. *IEEE 802.2, Logical Link Control.* New York: Wiley-Interscience, 1998.

—-. *IEEE 802.3, Local Area Networks and CSMA/CD Access Method.* New York: Wiley-Interscience, 2005.

—-. *IEEE 802.4, Local Area Networks: Token-Passing Bus Access Method*. New York: Wiley-Interscience, 1988.

Palmer, Michael J., and Bruce Sinclair. *A Guide to Designing and Implementing Local and Wide Area Networks.* Boston: Course Technology, 1999.

Stallings, William. *Local and Metropolitan Networks.* 6th ed. Upper Saddle River, NJ: Prentice-Hall, 2000.

Webopedia. http://webopedia.internet.com.

Wikipedia. "IEEE 802." http://en.wikipedia.org/wiki/Category:IEEE_802.

5 Network Software

Introduction

In this chapter we discuss network systems. The specific systems selected for discussion here were chosen because they made the best possible illustration of the variety of system types available. Their inclusion here does not constitute the author's endorsement of them. Much of the information about these systems cited here comes from the manufacturers' feature lists. However, the author has offered comments based on his experience (or lack thereof) with given systems. Certainly, the manufacturers' specifications and maintenance literature take precedence over any system data presented here. Given the unusually contentious atmosphere surrounding discussions of network operating systems and particular types of industrial control systems, the author hastens to add that the opinions expressed here are his own, not those of any organization, particularly the ISA, and that he has neither been paid nor will receive compensation of any type for his comments for or against any operating system.

Object-Oriented Programming

Much has been written about what object-oriented programs (OOP) are and are not. Their advantages include reduced development time, better organization of programming efforts, and reusable code. In fact, today any modern network system used in either commercial or industrial applications should be object oriented and have a good "object model" (the structure and interface requirements for classes and objects). OOP programs have classes of objects. A particular manifestation of a class is called an "instance" and to create such an instance is called "to instantiate." The concept of class can be explained by an example; a "class" of objects known as Dog. An instance of the class Dog is an object called Poodle; another instance of this class is called Doberman. They have the same main features found in all objects of the class Dog, but they are different. If you had an instance of the class Dog, you could make your own object by changing the various properties (size, weight, color, hair style, various appendage proportions)—hey, you could have a cocker spaniel. The point is that only the class must be defined, and then all the other objects can be made from it. This object orientedness enables software to be built out of components, rather than be all original or new each time. An automobile is built from components, so also a complex software program. This fact is the basis for the Component Object Model (COM) and now .NET, which are a legacy and current object model respectively.

To be robust, extensible, and scalable, a networked application should have a good object model. In computing, an object is a fragment of code and data that can be summoned or called (like a function) and may be reused, have siblings and children, and generally make life easier for programmers. Objects are also the basis for many other network application software programs such as Sun Microsystems' Java. Objects make possible faster production of programs, more consistency in program operation, less code, and the like. An object can be something as simple as the code for a push button. Using this code you could indicate whether the push button is depressed, has been depressed, or is not depressed. The object code could make the button illuminate, could dictate the alphanumeric on or around the button, locate the button's position on a screen, and determine its size, color, and illumination. All of these parameters are easily amended as attributes of a model and simple data entry (by manual or automated means) can set these values.

Objects can also be as complex as a chart recorder or a PID controller. Microsoft's Visual Basic is an example of a programming language that employs objects. Microsoft has advanced OOP along so that now a program can be built entirely of software components (objects) using a COM server (Microsoft Transaction Server for NT, Component Services for Windows 2000). Those familiar with Windows (2000 or XP) know that most applications are made up of an .exe program (the executable) and usually one or more DLLs (Dynamic Link Libraries). And DLLs are actually objects in the Component Object Model (COM). What Microsoft used to call object linking and embedding has essentially had its name changed to ActiveX . The difference between an .exe and a .dll is that the .exe is referred to as "in process"—same memory space; and the .dll is for "out of process"—running in a different memory space. You may purchase Microsoft's OCX controls (a set of ActiveX components including an executable) or write your own Java ActiveX objects, both of which may very well be a program or a set of programs in an object. Companies use object-oriented programming typically with Sun Microsystems' Java, which assumes that a Java virtual machine is on the device receiving the Java code. JavaScript, which is a scripting language, bears no relation to Java other than similarity in name (now called ECMA Script), and in a C++ programming style. It was developed by an entirely different company and then purchased by Sun Systems. Today, COM and Distributed COM (DCOM) are now "legacy" items. Taking their place, Microsoft now uses .NET, a system based in part on Extensible Markup Language (XML) and what was formerly called Simple Object Access Protocol (SOAP).

An entire set of objects called Object Linking and Embedding (OLE) for Process Control, or OPC, is used in many industrial applications today. We discuss OPC further toward the end of this chapter.

Commercial Systems

Though this is a book about industrial data communications, most of the systems used in the industrial environment had their start in the commercial world. At present, the world is divided into several camps, all of which give the impression that they are "standards" based. Unfortunately, interoperability is a bit difficult.

Stand-alone Systems

Not so long ago, the PC was much like that shown in figure 5-1: not networked and not connected to anything but the occasional bulletin board. Note how self-contained this so-called one-tier PC model was.

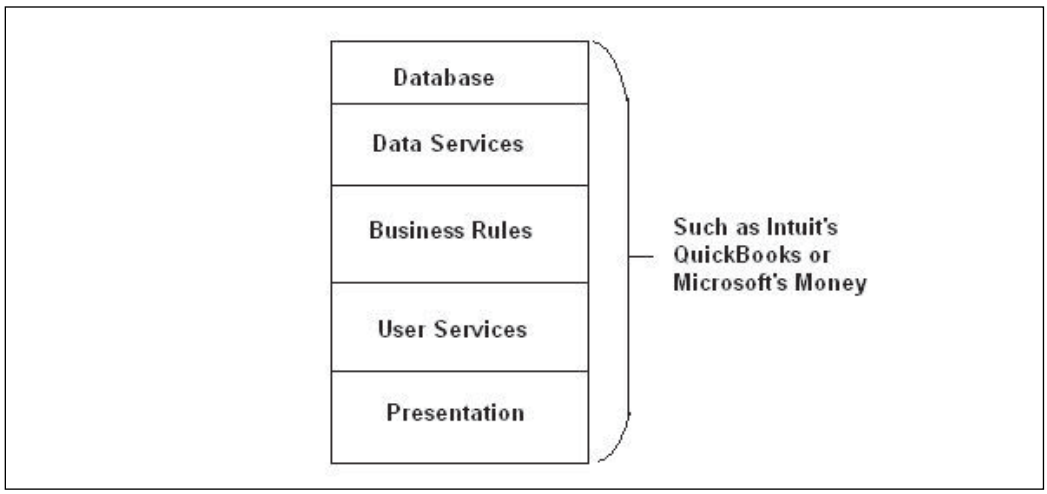


Figure 5-1. Self-Contained (One-Tier) Model

This one-tier model illustrates the way most PC application software was written. A proprietary presentation method (usually a graphical user interface, GUI) would present the information to the user. User services would interface between the business rules and the presentation, usually by presenting hooks (a programmer's way of saying addresses that handle events) to the resident operating system for using the GUI. The business rules would dictate what the user could and could not do with the data. Data services would interface the business rules and the database, which, of course, stores the information. This model would be very much the way a stand-alone analyzer in the industrial area would operate.

Two-tier models, such as client-server, are more complex. Before discussing the client-server model, we need to define these terms. Table 5-1 lists what a server shares. If a device shares a resource, it is a server. A client is a device that uses server resources.

Servers provide the sharing of resources such as:
printers
scanners
floppy/hard derives
cd-roms
tape backups
any shared peripheral
Servers provide network resources and network services such as:
communications
e-mail
Internet
Web
printer
database
backup
network management
security
file

Table 5-1. Server Resources

Two-Tier Systems

This is the so-called client-server model. Though it may reside on one machine, typically it will be found on two machines, one being a server (any device that shares resources) and a client (any device that uses the shared resources). In figure 5-2, you will notice two diagrams, one for the thick client and one for the thin client. The difference, of course, is where the business rules are located.

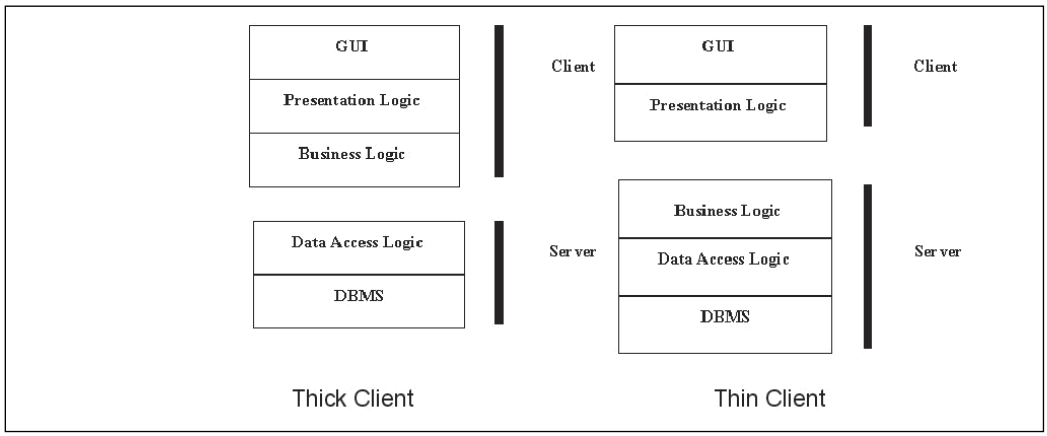


Figure 5-2. Two-Tier Systems

The advantage of the thick client is that if the database server is unavailable and this client is attached to another database server, then the business rules for this client are intact (hopefully, the database is a replicate of the one that failed; otherwise, problems will result). The thin client, on the other hand, is good for the administrator because there's only one place to upgrade and correct, and one point of control. The clients can actually be diskless machines that will not operate unless the network is up. Client-server is probably not the model to choose for industrial use though it has been used in that capacity.

The problem with the two-tier model is that it has no way of distributing functions. You have the client and you have the server—what more could you want? To distribute function we go to the three-tier or n-tier model. Actually, since the functions on a three-tier model may be spread out over three or more machines, the three-tier model is just a subset of the n-tier model. Figure 5-3 is a three-tier model, and figure 5-4 is an n-tier model.

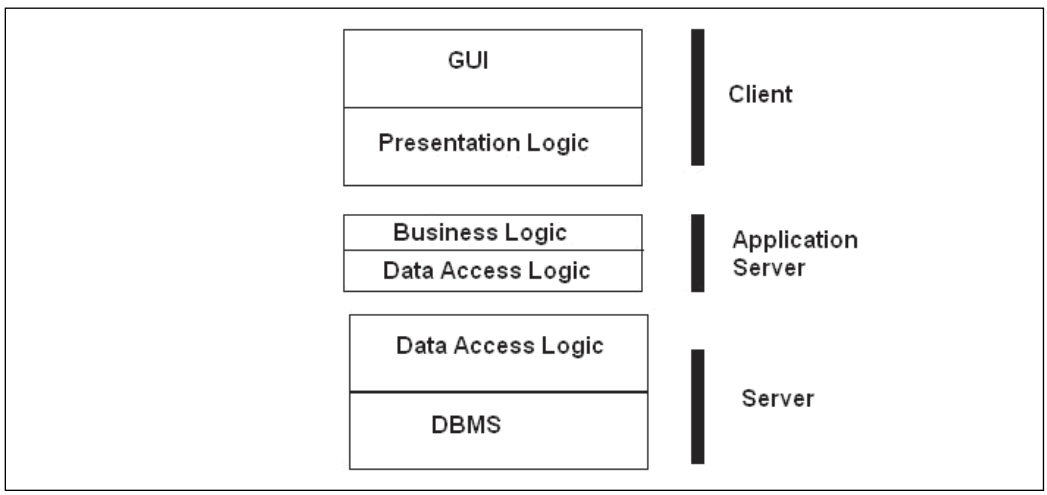


Figure 5-3. Three-Tier Model

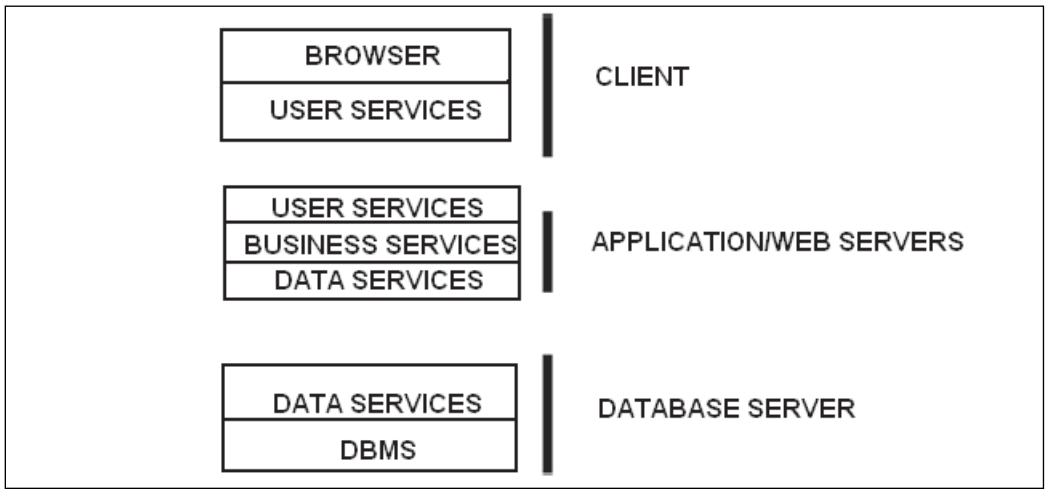


Figure 5-4. n-Tier Model

The n-tier model in figure 5-4 is using an Internet perspective. Certainly, the browser is an inexpensive display device. Traditionally, the control field has used the three-tier model for distributed control, using a sole-source proprietary system or mixed stores such as PLCs, somebody's database, and an integrator's HMI to put together a distributed control system. In commercial systems, the integration of the Internet with all business work has meant the widespread use of the n-tier model, which can be used effectively in control systems, even those not attached to the Internet.

One difference between the three-tier and n-tier models is how data is accessed. In the three-tier model, the client interface (typically a proprietary Visual Basic or C++ interface) is operated and requests and receives information through the business rules server. The business rules server may very well use a process like Microsoft's ADO (ActiveX Data Objects) that requests data and receives responses from the database server. The reason for the business rules is security and ease of use. The client is not supposed to be able to access the relational database (RDBMS) directly, but always through the business rules. For the n-tier model, the user services employ the Hypertext Markup Language (HTML). The client presentation services are typically Internet Explorer, Opera, Foxfire, or Navigator. The browser presentation language (HTML) is being supplemented by XML (eXtensible Markup Language), which conveys meaning about syntax rather than simply page layout. One of the present n-tier models has the client presentation services requesting responses (via HTML) such that one of the application Web servers will request data via an active server page (which is enforcing the business rules) from the database server. Receiving the data, the active server page puts the response into HTML and returns it to the client.

The n-tier model generally used today is called the "application model," which is illustrated in figure 5-5.

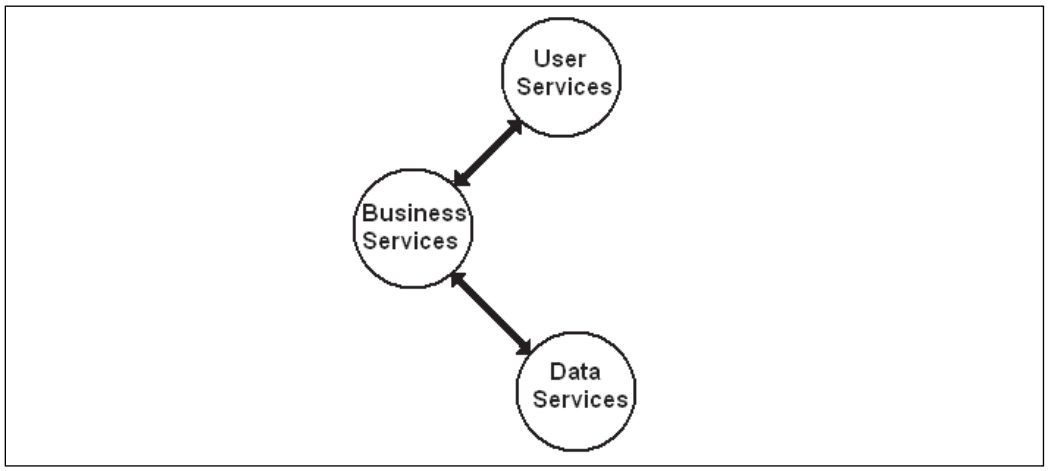


Figure 5-5. Application Model

Note that in no cases can the user services ever directly connect to the data services; this ensures that the business rules are enforced. The direct applicability of the application model to control systems should not be overlooked.

The Internet

As a networked system, the Internet is an n-tier design. We describe it in some detail in chapter 8 on Internetworking, but the key point here is that both it and the technologies it has spawned are either used now or soon will be in your industrial systems. The application model and Internet n-tier technologies have already revolutionized office systems by coming closer to the goal of providing simple, universal, and platform-independent access. As a medium for connecting geographically separated plants the Internet offers the tremendous advantage of costing only the services of an ISP for truly high-speed interconnection. *Please note here that the author does not advocate running a control system over the Internet; however, Internet technologies have established themselves on company Intranets—the facility network, not the Internet.*

Network Operating Systems

It would seem at this point, that logic dictates a discussion on the basic operating programs known as the Operating System (or OS). These differ from applications programs in that the OS supplies the basic services for the applications. An entire book could be devoted to operating systems, their design and implementation. There is neither the space nor the requirement for a very detailed explanation so only some salient facts (affecting applications) will be presented and only on a few of the more popular operating systems. If your favorite system is not included, my apologies.

Microsoft Windows

Microsoft Windows® has the largest market share of desktop systems (over 90%) and is making significant inroads into the server business (particularly in process control). It acquired

its share by offering reasonably priced software that would do 90% of what most people wanted at least 90 percent of the time, and with an easy-to-use interface. No one who has ever used the Windows 9X series of OS (95/98/ME) will say that it always did what he or she wanted. But it certainly gave reasonable performance for its intended use and price.

Microsoft offers a myriad of operating systems under the title "Windows," which are actually different in concept and execution. Windows 3.1 (barely used anymore) was a 16-bit system with a GUI glued on to Microsoft DOS. Windows 95 was a hybrid system supporting 16-bit DOS programs (using shared memory, cooperative multitasking) and true 32-bit (Win32) programs (protected memory and preemptive multitasking). As with anything, trying to be all things to all programs meant that it did not excel at any one of them. Windows 98 and Windows ME are really upgrades to 95. Windows 98 added support for "plug and play," and ME is multimedia oriented.

Windows NT ("New Technology") was good to start with (protected memory, small kernel, preemptive multitasking) and has done nothing but improve. Windows 2000 added plug and play and less rebooting was required to refresh the registry. W2K Professional was a suitable industrial workstation OS for almost any application, and the server editions (Server, Advanced Server, and Data Center Server) performed reliably in the most trying situations. Windows XP was an effort by Microsoft to do two things. First, it wanted to establish a standard code base across all of its operating systems (no more "9X"), XP Home, XP Professional, and Server 2003. Second, it wanted to move more into subscription services, where software is rented rather than purchased in perpetuity (at least that was the thought at the time; it seems it has not quite worked out that way yet). It is a matter of fact, however, that Windows XP Pro (Service Pack 2—with the seventy-nine (6/2007) security patches) is quite probably one of the best clients available (my opinion, remember?).

Windows CE is Microsoft's entry into the small-footprint operating systems' market for personal digital assistants (PDA) and sub-notebooks, including tablets. It has found application in industry because of its low resource requirements (compared to its big brothers), while still interfacing easily with its kin. Microsoft has made a big marketing push to have Windows CE or a selective combination of CE objects on wireless phones, copiers, and office equipment of any description.

Windows CE (5.0) became Windows Embedded CE 6.0, (12/2006) a componentized operating system designed to power small footprint devices, such as:

- set-top boxes
- thin clients
- digital media adapters
- voice-over-IP (VoIP) phones
- navigation devices
- medical devices

- portable media players
- home gateways
- digital cameras
- networked digital televisions
- PDAs

Additionally, Microsoft has expanded its "embedded" family to include a version of XP. Windows XP Embedded with Service Pack 2 (SP2) is the Windows operating system in componentized form for rapidly building reliable and advanced embedded devices. Based on the same binaries as Microsoft Windows XP Professional, Windows XP Embedded enables developers to pick and choose individual components in order to achieve optimum functions in a smaller footprint. The operating system software also provides the latest multimedia and Web browsing capabilities.

And new to the scene (as of 2007): Microsoft "Vista." Vista (Longhorn previously for servers) has a number of features eminently useful for the industrial arena, such as specifically allowing users to operate many programs (particularly legacy programs) without being an administrator. If you have ever struggled with determining permissions across one or more Windows domains, Vista is the answer. Additionally, even its deployment is easier because it uses a file format for disk imaging rather than a sector image. This means that many tasks, such as adding drivers, can be done without rebuilding the system image, and so on.

UNIX

UNIX, which has been around for over forty years, has had time to mature. Yet because vendors' products—Sun's Solaris, IBM's AIX, Apple's OS X, SCO UNIX, et alia—needed to modify the kernel to do specific tasks they deemed necessary, UNIX became a multi-flavored operating system that in many cases was not source-code compatible. UNIX vendors have partly consolidated and partnered in the past few years, yet the bulk of their competition comes from a UNIX clone, Linux. UNIX has suffered through the years from each vendor's lack of sufficient market share to mass-market (think: shrink-wrap) the software that runs on the system. Hence, it was (and is) more expensive than the Windows equivalent. To make matters worse (from a marketing standpoint) UNIX was a command-line (not GUI) based system. Though many front ends have been applied to UNIX to give it a graphical user interface, they do not have the low cost or popularity of Microsoft Windows. At the risk of getting mired in the religious wars of operating systems; and I make these comments as a user and programmer (and elder observer), I believe that grown men should be looking at something other than a particular operating system as the devout cause of their life. UNIX has the history, stability, and extensibility to remain in use in many high-end systems, but cost and the improving Microsoft and Linux products have challenged UNIX even at that end. This is particularly the case in industrial applications, where the tools, ease of use, and cost of Windows NT/2000/2003 and Linux have greatly eroded what was once a UNIX-only bastion.

Linux

Linux is an "open-source" clone of UNIX (Free BSD is open-source UNIX). Open source doesn't necessarily mean free, only that the source code is available. Linux has a good reputation as an application server, and its total cost of ownership rivals or, in some cases, betters that of Windows. It is in the application software field that Linux has had problems. Though there are many good Linux programs that will do at least some of what you want on Linux, most programming firms do not want to give away software or intellectual property rights, which they must by virtue of the open-source license agreement (these firms are actually in business to make money from programming services). As a result, Linux applications are somewhat limited. Many of the programs that are available may be challenging for a nonprogrammer to install and difficult certainly for the average operator, particularly when compared with Windows' ease of use.

However, with each subsequent release, Linux closes the gap. A number of quite usable GUIs are available for Linux, and now the major competitive issue in the open-source camp is whose distribution is better. Sun has thrown its weight behind Red Hat, while recently Microsoft (yes, you read that correctly) is assisting with a competitive distribution; Novell's SUSE. Apparently, Novell will do the open-source things while Microsoft will make Windows (particularly the servers) work better with Linux—a win-win for users, if it works out.

Other companies make network operating systems (e.g., Apple, which uses UNIX as the core of it operating system), but Windows, UNIX, and Linux are the current big three. Naturally, any company in the industrial networking business would like to build off of an already successful system, as writing an operating system is difficult and tedious at best, particularly when graphics and interfaces to other systems are involved. Yet most industrial system vendors did just that, they designed, programmed and implemented their own operating system, usually a derivative of UNIX, having to develop all the applications, interface screens, graphics. Some still try—see chapter 6 on industrial networking and fieldbuses. In that chapter, you will find some vendors still trying to reinvent the wheel and others capitalizing on successful operating systems.

Windows, UNIX, and Linux manage Layers 5 through 7 somewhat differently. Most layers handle communications in a protocol stack, but Layers 5 through 7 are bound (by a process referred to as binding) from the stack to addresses and locations that are generally part and parcel of the operating system. All three operating systems claim to be compatible with network "standards," and all have differing degrees of compliance. All offer many features not needed in an industrial operating system and leave out a few that are required. Perhaps the most missed feature is a real-time dispatcher that would limit the uncertainty over execution times (but that in itself is also a problem with the asynchronous PC).

Protocols Used by Vendors

Each vendor has a preferred set of protocols. Each will supply the TCP/IP protocol stack and generally the entire suite of protocols, so we will discuss others first. Windows NT (in addition to NetBUEI) supplied its own implementation of the other vendors' protocols, such as Apple's AppleTalk, Netware's IPX/SPX, and IBM's DLC. Windows 2000 supports other protocols but defaults to TCP/IP, while XP can use others but is designed for TCP/IP (which is required if one is going to run Active Directory services). Novell (version 5.2) offers TCP/IP as its native connectivity, and Apple's OS X is actually a flavor of UNIX, so it offers TCP/IP as a native protocol.

Microsoft's NetBEUI

NetBEUI stands for "NetBIOS Extended User Interface." NetBIOS is not a protocol but an application programming interface, a naming convention, and a way of interfacing network hardware and network software. NetBEUI handles the transport of this data. For small networks (less than one hundred nodes), NetBEUI was as close as one could once come to a plug-and-play network. There is very little setup—you only have to assign a unique name to each piece of equipment on the network. Identical names are not allowed as they would confuse the protocol. Using each node's unique computer name, NetBEUI is actually a Layer 4 function providing for end-user packet sequencing and recovery and interfacing with the LLC of Layer 2. What is missing here is Layer 3. NetBEUI is not routable. It has no provision for routing (it was intended for small networks) and is too limited for WAN use because of its use of acknowledgment windows. To route NetBEUI (it can be done) it must be encapsulated in either an IP or IPX packet. However, the interfacing ends have to be "spoofed" or the time to respond will expire on the LANs. NetBEUI is a legacy networking protocol and there is no justification for its use (unless you have a legacy network) as TCP/IP networks are now just as plug and play typically as NetBEUI ever was. And remember that almost all the network diagnostic and troubleshooting tools came from one vendor; very few firms marketed to this segment.

However, Windows networking still requires the computer name be unique (as well as IP and MAC address). Table 5-2 lists the major protocols unique to Windows networks.

Abbrv.	Name	OSI Layer	Brief Description
SMB	Server Message Block	7	Opens/closes files, reads/writes data blocks to open files, lists directory, provides UNC names, manipulates registry entries (server), and provides high-level connection services.
NBT	NetBEUI	4-6	NetBEUI over TCP/IP using the NetBIOS API
NBF	NetBEUI	4-6	Network BIOS Extended User Interface (a protocol) uses the NetBIOS API
None	NetBIOS	2-5	Network Basic Input/Output System (an API)

Table 5-2. Windows OSI Layer Implementation

All Windows' nodes use Microsoft networking. Part of Layer 7 is the Server Message Block (SMB). It uses the NetBIOS computer name. Microsoft states that the name must be in upper-case letters and can be up to fifteen characters in length. There are a number of variants of the SMB so the first transmitted block will contain the "negprot," which negotiates SMB features. Some of the variants include particular SMBs for Windows NT/2000/2003 and for Samba which is an interface between Windows networking and UNIX computers (i.e., using Samba a Unix computer can appear on a Windows network and communicate. SAMBA allows a Unix system to support SMB and let a Unix system appear in a Windows network neighborhood and share its file system).

CIFS: Common Internet File System

The TCP/IP suite offers FTP for file transfers, but it is somewhat limited when it comes to file sharing for applications and browsers. The Common Internet File System (CIFS) is a variant of the SMB that Microsoft proposed to the Internet Engineering Task Force (IETF) as an open standard. Though some non-Microsoft people have complained that CIFS is not as open as they would like, a surprisingly large number of non-Microsoft SMB variants are available. Since Microsoft-based clients are a large number of the Internet hosts (nodes) it was highly likely that CIFS would reign and that Sun's NFS (Network File System—the UNIX version of SMB) be used strictly for UNIX-only answers.

Netware's IPX/SPX Suite

Until version 5.0 and up, Novell used its own IPX, derived from Xerox's XNS protocols (see table 5-3). Note that to talk to Netware from Windows requires that you have a device (or software programming) that translates the different Layer 7 protocols even if both are running IPX/SPX. This is only a historical reference as IPX is now legacy software.

Abbrv.	Name	OSI Layer	Brief Description
SAP	Service Advertising Protocol	5-7	Allows clients to identify servers and network services
NCP	Netware Core Protocol	5-7	Opens/closes files, reads/writes data blocks to open files, lists directory, and provides high-level connection services.
SPX	Sequenced Packet Exchange	4	Transport for connection-oriented applications
IPX	Internet packet Exchange	3	Routing protocol
RIP	Router Information Protocol	3	Provides routing information about networks to routers

Table 5-3. Novell OSI Layer Implementation

TCP/ICP Suite

There is no UNIX suite since UNIX has for the most part used the TCP/IP suite. Sun originated and most UNIX machines use the Network File System. This is a Layer 7 protocol and performs the same functions as the Microsoft SMB and the Novell NCP. Though there are other programs in the suite, table 5-4 lists the most common Layer 3 and above protocols.

Abbrv.	Name	OSI Layer	Brief Description
HTTP	Hypertext Transfer Protocol	6-7	Used in Web communications
SNMP	Simple Network Management Protocol	6-7	Used to manage network nodes and devices
FTP	File Transfer Protocol	5-7	Transfers files from one network location to another
Telnet	Telecommunications Network	5-7	Has a workstation emulate a dumb terminal
SMTP	Simple Mail Transport Protocol	6	A "standard" for e-mail
RPC	Remote Procedure Call	5	Enables a computer to execute services on a remote computer (usually a server)
TCP	Transport Control Protocol	4	Ensures end-user-to-end-user data reliability
UDP	User Datagram Protocol	4	A connectionless service with no acknowledgment
DNS	Domain Naming Service	4	A distributed database of IP to computer names
IP	Internet Protocol	3	Routing protocol
ICMP	Internet Control Message Protocol	3	Network error reporting
RIP	Routing Information Protocol	3	Provides routing information about networks to routers
OSPF	Open Shortest Path First	3	Provides routing information about networks to routers
ARP	Address Resolution Protocol	3	Resolves IP addresses to network name address

Table 5-4. Internet OSI Layer Implementation

It should be apparent that if you use standard Layer 1 through 4 protocols then the only differences are really in Layers 5 through 7. For industrial applications with the three-layer model (1, 2, and 7) there is a similar problem. Even though the different network operating systems have a common standards-based set of protocols, they still cannot talk directly to each other. This is illustrated in figure 5-6. Although I took a little liberty with the layers that SMB, NCP, and the TCP/IP upper-layer protocols actually cover, I am sure figure 5-6 leaves no doubt about the current problems posed by using mixed platforms.

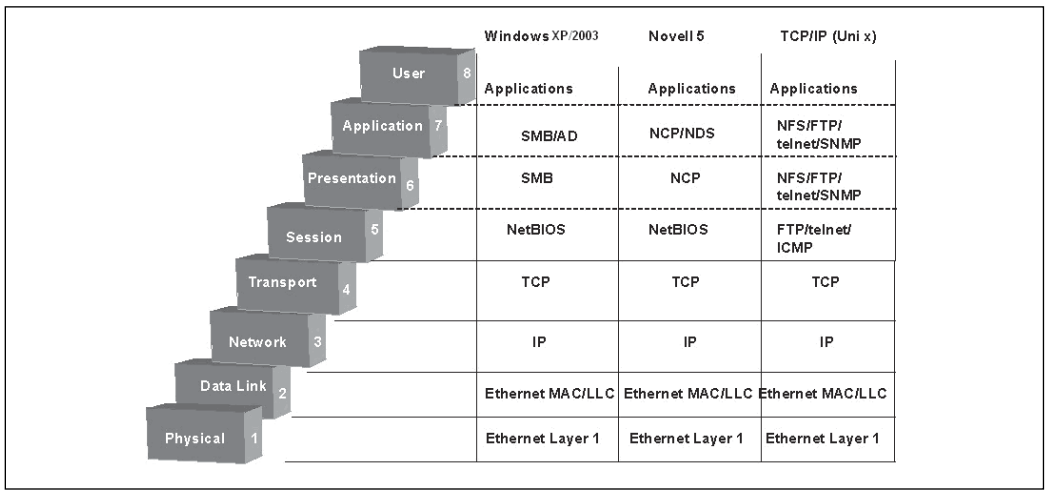


Figure 5-6. Network Operating Systems and the OSI model

An Application Object Model: OPC

Object linking and embedding (OLE) for process control (OPC) is currently based on Microsoft's COM (Component Object Model) and DCOM (Distributed Component Object Model); however, as COM and DCOM are legacy protocols, OPC is transitioning into the OPC-UA for Unified Architecture, an open (based on network standards) system. The object set started out being called OLE and is now under the umbrella name of ActiveX technologies. OPC consists of a standard set of interfaces, properties, and methods for use in the process control industries. It is analogous to having software components likened to the instruments and OPC likened to the 4-20 mA loop. COM itself is a binary standard, meaning it is generic and not beholden to any particular development language. COM will run on platforms other than Microsoft (however, Microsoft now considers it to be legacy code). The goal of OPC is "plug and play" for process control where only one set of drivers for a device has to be written and may be reused, where only one software toolkit is required for development, and where the configuring of software and hardware is automatic. For example, a PLC may perform data collection from its I/O, and then it will become an OPC server for those OPC clients. Application developers can write in whatever language they deem appropriate. OPC offers a number of benefits to users:

- lower system integration costs
- ease of integration (plug and play)

- auto-configuration of tags
- elimination of proprietary lock
- access to data by every level of the hierarchy

Figure 5-7 illustrates the concept of OPC.

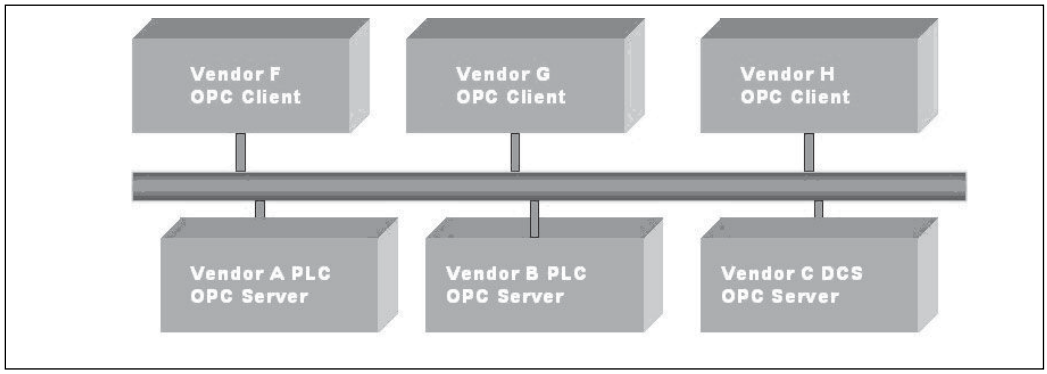


Figure 5-7. OPC Concepts

Other applications are pushed by this organization or that organization. However, OPC obtained a critical mass of users and vendors and was one possible answer to the "Whose doing Layer 7?" problem, since OPC runs on top of Layer 7 (whoever's) and implements the application model described previously. The OPC Foundation has support from many automation integrators and suppliers. Microsoft has hosted OPC Foundation's annual meeting but has otherwise kept a very low profile, supplying technical support and briefings on software upgrades that might affect OPC. The COM model and OPC will run on non-Microsoft platforms.

Things change. OPC now offers a unified architecture (UA) that is designed to use open standards and those OPC programming models that are currently employed. This unified architecture is developed in a layered manner so technology changes will not require rewrites (such as switching from COM). It is also designed to be forward-looking and backward compatible, preserving users' investments in the previous issues of OPC.

Three different code bases will be used:

- *ANSI C/C+* meets specific requirements for embedded systems.
- A portable Java implementation,
- *Microsoft .NET* where the Microsoft .NET 3.0 platform will be used.

The OPC Foundation has established a time frame for delivering code that "wraps" previous implementations of OPC, making it backward compatible, and also a schedule for features new to OPC as a result of the unified architecture. Microsoft's .NET has delivered the most complete features set, but then .NET itself is capable of being run on many different architectures and operating systems because it is standards based. While some UNIX

flavors will run COM/DCOM (reluctantly), however, that first incarnation of OPC was essentially a Windows-based suite; the Unified Architecture overcomes that limiting feature and runs on many architectures.

Conclusions

As we have seen, communicating between dissimilar platforms is almost as confusing in the commercial world as in the industrial one. And the problems the two spheres share will soon grow when most industrial networks standardize around Ethernet (Layer 1 and 2) and TCP/IP (Layer 3 and 4), leaving the operating system to determine Layers 5 through 7. If you wished to communicate between different platforms using LAN technologies, somewhere along the line you would find a gateway—a computer with two network protocol stacks to convert between this Layer 7 and that Layer 7.

Yet, if you will recall, the n-tier method wasn't terribly dependent upon platform, only upon the capabilities of the presentation software (the browser). After all, a browser looking for a Web site generally can't tell if the site is running on Microsoft IIS (over Windows 2003) or on an Apache Web Server (over Linux). As XML is used much more widely, platforms will not matter as much as the presentation program. XML is capable of interprocessor communications, particularly when coupled with Simple Object Access Protocol (SOAP). Whether running on the Internet or over an intranet, the techniques are the same. It should be noted that though Microsoft Office 2000 (a widely used suite of productivity tools including word processor and spreadsheet) has some XML enablements, Office 2003 (or 2005 depending upon feature delivery) is totally XML enabled, while Office 2007 (current version) offers even more standards-based features. However, though Office 2007 comes in a multitude of feature formats to be cost effective for its target audiences, its core is the same. It is a fact that other programs vying for the same market space will be XML enabled also.

Summary

This chapter discussed some of the models used in commercial networks as well as some of the actual vendor products used in commercial networks. Since both the feature set and details are apt to change during this book's life span, the level of detail provided here is sufficient. In fact, this chapter could be summed up as follows. For commercial network software, Layers 1 and 2 are Ethernet, Layers 3 and 4 are TCP/IP, and whichever operating system you choose will supply Layers 5, 6, and 7. You will be using either the two-tier, three-tier, or n-tier model, and your application software will run at the user layer.

Bibliography

Note that Internet links may change.

Microsoft Corp. *MSDN Library.* Microsoft, July 2000. (corporate Website) http://www.microsoft.com.

Morneau. Keith. *MCSD Guide to Microsoft Solution Architectures.* Boston: Course Technology. Keith, 1999.

Novell Inc. (corporate Website). http://www.novell.com/linux.

OPC Foundation (organization Website). http://www.opcfoundation.org.

Palmer, Michael J., and Bruce Sinclair. *A Guide to Designing and Implementing and Local and Wide Area Networks*. Boston: Course Technology, 1999.

Red Hat Inc. (corporate Website) http://www.redhat.com.

6
Industrial Networks and Fieldbuses

We start off this chapter with much the same disclaimer as the previous one. Of the numerous industrial systems available, the author has selected for discussion those that illustrate a type, have a large installed base, and/or offered the author sufficient technical information (as opposed to marketing features). The twelve or so selected here are representative of many. Again, the author accepted no payments, fees, or quid pro quos for describing one vendor's product over another's. And as always the author's views and descriptions here are his own and do not represent official endorsement by the ISA or anyone but the author.

The Many

Before launching into the requirements of an industrial network, a word of caution is in order. For just fieldbuses alone, a simple perusal of the Synergetic Fieldbus Comparison Chart (www.synergetic.com) lists eighteen (count 'em) different fieldbuses, even excluding two types approved in IEC 61158. IEC 61158 is the international (read: European) fieldbus standard that standardizes around eight non-interoperable protocols. If you perform an Internet search of "industrial Fieldbus" you will find over 138 different protocols and systems are used in industrial applications. Many entries will be user groups, and many point to one or another of the popular protocols. It should be obvious that we do not have the space to explain each and every one in this chapter. So we have selected those that best illustrate key principles.

Industrial Network Requirements

All networks, commercial and industrial, share some common requirements:
They provide effective performance for resources used

- They offer multilevel security
- They are cost effective
- They are standards based
- They provide reliable transmission
- They offer ease of access
- They provide ease of use

In addition, industrial networks have these extra requirements:

- They must offer predictable throughput
- They must provide predictable scheduling
- Their downtime rates must be extremely low (nonexistent is preferred)

- They must be able to operate in hostile (to equipment) environments
- They must be scalable from one to many
- They must be operable by other than communication specialists
- They must be maintainable by other than communication specialists

All designs are compromises, however, so some of the common requirements (such as being standards based or cost effective) may have to give way to ensure key industrial requirements such as low downtime, and so on. Let's take a closer look at the seven special requirements of industrial networks.

Predictable Throughput
Industrial requirements are such that response times must fall within a specific window of time or an error can result. This functionality is called determinism. Generally, the window must be wide enough so that the response can be acted on in real time. That is, the controlling device must issue an output in time to affect the control or alarm before a processes' operation becomes unstable.

Predictable Scheduling
Time-sensitive operations such as computing a PID algorithm must provide a definitive window of time for data input or the results will be incorrect.

Extremely Low Downtime
If the control system is down, the process is down and the company is losing money. Downtime requirements in modern industry are such that a system must be up above the "five nines" (99.999%), so downtime must be less than 0.001 percent of total system availability. This may only be achieved by using distributed control, redundancy, fail-over clustering, or other techniques that will reduce system nonavailability under any circumstances.

Operation in Hostile Environments
Industrial operations are usually performed in extremes of temperature, vibration, chemical atmosphere, electrical interference, and lack of cleanliness. Only commercial equipment that has been industrially hardened or encased in an industrial enclosure can withstand this treatment (when is the last time you hosed down the wiring closet?). Fortunately, much of the data communications equipment in industrial settings can be located in the control room, where the environment is more closely monitored and maintained for human activities.

Scalable from One to Many
Scalable (also spelled "scaleable") means the design doesn't lose its efficiency when the network is expanded by a magnitude or more. Though additional resources will be required, the overall design should accommodate very small to very large network configurations with no loss in features and timing.

Operable by Nonspecialists
Most operators know their process, but asking them to learn the intricacies of communications is probably beyond their desires and training. In an appropriate industrial control system the communications technology will be totally transparent to the operator. He or she shouldn't have to determine anything networking related except how to report an alarm to the correct party.

Maintainable by Nonspecialists
Maintaining industrial control systems has become a very involved process. The days of electronic component repair are gone. Now, locating the problem in a unit requires programming, diagnosing, and analyzing systems. If the problem is communications related, the process control technician generally does not have the background, training, equipment, or skills to quickly and efficiently locate it. A well-designed system will have sufficient diagnostics such that a control technician can find the errored unit and replace it without having to suffer through ROM revision levels, protocol switch settings, and missing software components.

In many cases, the effort required just to meet the last two requirements drives an industrial system's cost far above a commercial system offering similar performance. One method developed to address these specialized requirements is to provide a proprietary turnkey approach—the vendor-proprietary distributed control system (DCS). This has evolved into the open DCS.

Distributed Control Systems

In distributed control systems, the intelligence is located in various nodes that are connected by some media and networked together, enabling the nodes to perform peer-to-peer communications and for each node to perform a portion of the overall set of tasks. For several years the DCS area was the province of proprietary systems. The amount of capital needed to develop a system, work out the bugs, and market, install, and provision such a system usually necessitated a fairly well-established company. The number of potential customers would be few, and so the cost could not very well be amortized, resulting in relatively high prices for these systems.

At the low end (cost wise), the programmable logic controller (PLC), a device developed to replace relay logic, enjoyed significant success in industrial applications. Here was a digital logic device that, as it grew, possessed increasing intelligence in various nodes, was connected by a data highway (really a local area network), performed peer-to-peer communications, and performed a portion of the overall tasks at each node. It was not called a DCS, however, for DCS had come to mean devices capable of performing continuous process control actions rather than just on/off or two-state control. Then analog I/O and the PID control strategy were added to the PLC. Now what? You may consider a facility of networked PLCs using a third-party HMI and performing both discrete and analog control to be a DCS. The author does, and so should most people. The difference is just

that the PLC/GUI setup will cost considerably less in initial costs (using market-available devices with their much lower research and development recovery costs means lower price).

Though PLCs are by and large proprietary devices, the Physical layer on most PLC networks are variations on standards. PLCs are used primarily for discrete manufacturing, offering only the PID strategy for handling continuous processes. Process DCS, on the other hand, will offer advanced strategies. Other than the Physical layer, PLCs have undergone very little standardization. Industrial communications—all three layers—must meet many rigorous requirements, most of which we have encountered already. It's a sound assumption that proprietary systems meet or exceed these same demands. It remains to be seen how the proprietary-versus-open networks battle will turn out; at present, open appears to be the encompassing wave.

Proprietary DCS: A Brief Look

Most instrument manufacturers also sell a distributed processing system. Typically, these systems perform PID control for several loops and offer advanced control strategies such as cascade, feed forward, ratio, and other multivariable control strategies. These proprietary distributed processing systems also supply trending, optimization, very nice graphics, diagnostics, operator stations, and an engineering station for configurations and changes. Originally, their communications methods were closely held. The first ones, like the Honeywell TDC2000 and EMC Controls Emcon-D and D/3, had totally proprietary operating systems in both the 'workstations' and 'operator consoles' and particularly in the distributed controllers. It was the next generation of DCS products, in the mid-80s where systems based on UNIX (one flavor or another) and using some form of serial communications begin to appear.[1] Typically, this was a token-passing bus or modified master/slave arrangement, connected as a logical ring that met the manufacturer's performance standards. These proprietary distributed processing systems were turnkey operations. Once it had been selected, you were locked in—at least for major components (and a great number of the minor ones too).

And what did one use as an external interface for these proprietary distributed processing systems? When provided, it would normally be compliant with either EIA 232 or 422. And the software for communications would typically be proprietary. Just having a standard communications port on a processor or two does not make for an open architecture, however. Time has passed, and many of those proprietary systems that remained closed are neither being purchased nor added on to. They are being replaced by either more "open" systems or by standards-based systems.

Those vendors interested in making these systems had to take several actions to remain competitive. First, they had to become truly more open. For a vendor this is a dangerous step, because the system and all its components are vulnerable to shifts in marketplace preferences. One method vendors undertook was to base their systems on widely accepted

platforms. Many of the old-line DCS manufacturers now base their systems on Microsoft Windows 2000/XP, while still offering support for their older UNIX wares. Microsoft Windows networking offers the advantage of a native GUI (not one added on or developed through many man-hours of programming), an excellent security model (though minor implementation bugs have occurred, each patch makes it a little less vulnerable), and reduced development costs while providing an easy interface to office systems software. Having a host of third-party vendors offering components at any level allows for quick and reliable development.

Another vendor method has been to embrace the standards, by offering Ethernet and TCP/IP to the outside world and using various "standard" buses and protocols internally. The DCS manufacturer has in effect become a systems integrator even though part of the systems integrated are designed, developed, and manufactured by the integrator itself.

Programmable Logic Controllers (PLC)

The programmable controller using the analog I/O and a PID algorithm performs continuous process control and as such is a loop controller. Also offered is the PLC mainstay: discrete control. PLCs vary widely in their capabilities, and this is certainly not the place to list them all. Of interest is the communications between and with the PLC. Each PLC vendor has his or her own preferred scheme for communicating with most PLCs, but two predominate. Allen-Bradley PLCs use the DataHighway Plus (a proprietary scheme), and almost every other PLC vendor would accommodate Modicon's ModBus RTU (Remote Terminal Unit, a public domain protocol). We will discuss both in the following section.

Selected Industrial Systems: Allen-Bradley and Modicon

The concepts behind Allen-Bradley's DataHighway, DataHighway Plus, and DataHighway 485™ are illustrated in figure 6-1. The DataHighway is used for peer-to-peer transmission between controllers and is a local area network for up to 255 nodes (Octal 377, the 256th node, cannot be used as an address in that it is all 1's). The DataHighway bus operates at 57.6 Kbps and can have a trunk line of up to ten thousand feet (3.048 kilometers) at the 57.6 Kbps data rate, with drop lines from the trunk of no more than one hundred feet. DataHighway uses the modified token-passing scheme known as "floating master," which provides a bid scheme for controllers who wish to be master, in which they bid for the bus based on their priority and need to use the bus.

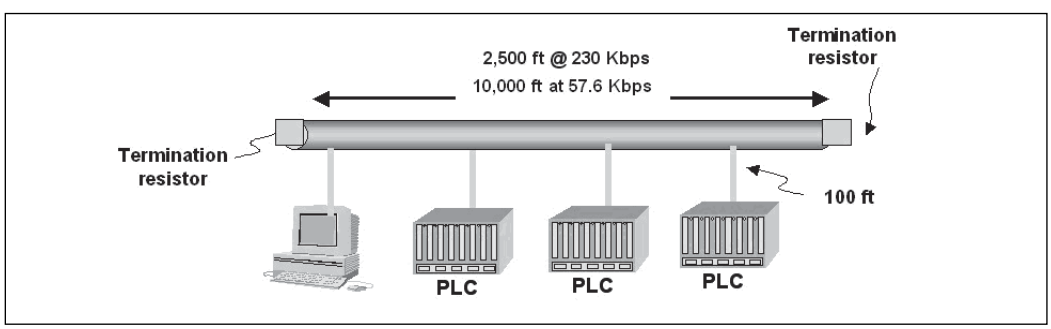


Figure 6-1. DataHighway/DataHighway Plus Topology

DataHighway Plus is a token-passing bus that allows up to sixty-four nodes (0-77 Octal) on one highway, with speeds up to 230 Kbps. This is not a field device connection, but a processor-to-processor connection for PLC2, 3, and 5 series devices. DataHighway Plus allows a PC (with a KT card and appropriate software) to be used as a programming or operator terminal. The primary differences between the DataHighway and the DataHighway Plus versions are that the latter was optimized for the lesser number of nodes and allows online programming. DataHighway uses ±14 volts peak-to-peak signaling, while DataHighway Plus uses ±7 volts signaling. Both use Manchester encoding.

The Data Link layer frame contains the flags and a two-byte CRC. The message structure is assembled by the Application layer software and then is encapsulated in the Data Link layer frame. Figure 6-2 illustrates how the three-layer (1, 2, and 7) is implemented using the DataHighway. Note that the Data Link layer is divided into the IEEE MAC and LLC.

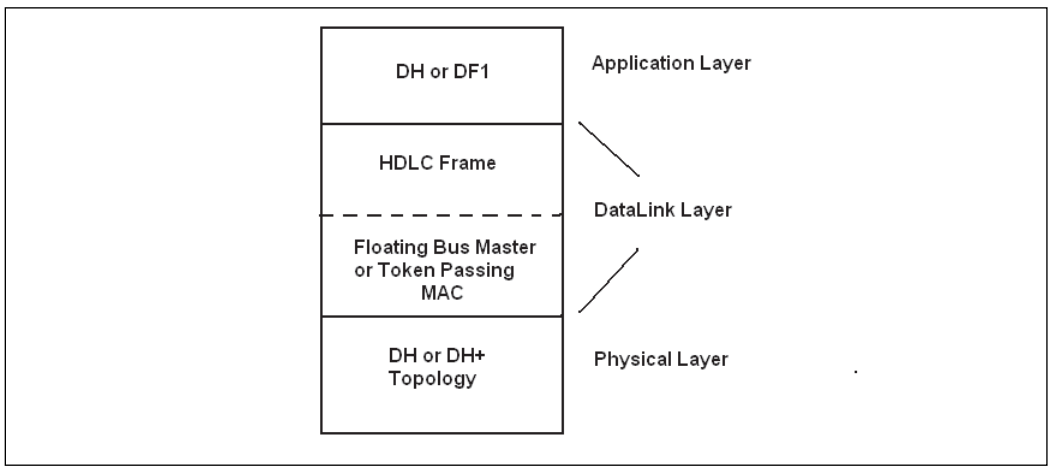


Figure 6-2. DataHighway/DataHighway Plus Layers

Figure 6-3 illustrates the message format for DataHighway and DataHighway Plus. The Application layer assembles the message (DST Address through Data), and the Data Link layer computes the CRCC, adds it to the frame, and frames the Packet Data Unit (PDU) with the start and stop flags. Zero insertion is used to ensure that there are no more than five zeros in a row unless the Data Link layer decides it is time for a flag. This is a synchronous protocol, meaning that system timing is dependent upon bit transitions and each bit must be accounted for. At the receive end, the flags are used to determine the frame length, and the CRCC is computed independently on the message and compared to the transmitted CRCC. If a match is made, the PDU has successfully been received; if there is no match then error correction (retransmission of errored packet) is required.

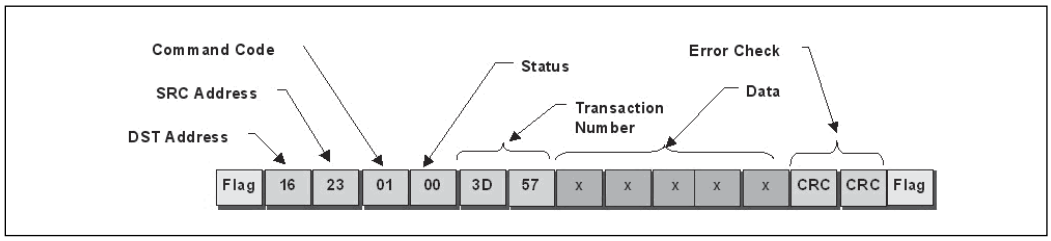


Figure 6-3. Message Format

DF1 is Allen-Bradley's asynchronous protocol that is transmission of one block following another may be at the next bit time or several hours later; data is said to occur at non-synchronous times. It is byte oriented rather than bit oriented and uses ASCII control characters (actually the control structure is quite similar to IBM's Bi-Sync). All control characters are preceded by a Data Link Escape (DLE) character. Some of the controls are as follows:

DLE STX	Start of text frame
DLE STX	End of text frame
DLE ACK	Positive acknowledgement
DLE NAK	Negative acknowledgement
DLE DLE	Data (which may have embedded control codes)

The DataHighway 485 uses the EIA 485 standard and will support up to thirty-two nodes. It has a data rate of up to 115 Kbps (although the EIA 485 maximum is 10 Mbps for fifty feet). EIA 485 refers to the electrical characteristics rather than how the network is set up. The network setup will be according to Allen-Bradley's definition of full duplex (DF1) or half duplex (polled), where (logically) the DF1 consists of two two-way paths and the half duplex of one two-way path.

Modbus RTU

The transmission method used by most other PLC manufacturers, in addition to their own proprietary method, is the Modbus Remote Terminal Unit (RTU). Its OSI implementation is illustrated in figure 6-4. This is a master/slave arrangement in which there is one master and one or more slaves. It is a query-response protocol. That is, only the master can query; the slave cannot initiate, only respond—a method similar to polling. Many manufacturers use Modbus RTU as an accepted standard, and the protocol can perform the basic host-to-PLC communications. Modbus RTU is not for a field device, but rather from a processor to a host, and it actually has two modes, an ASCII mode and the RTU mode.

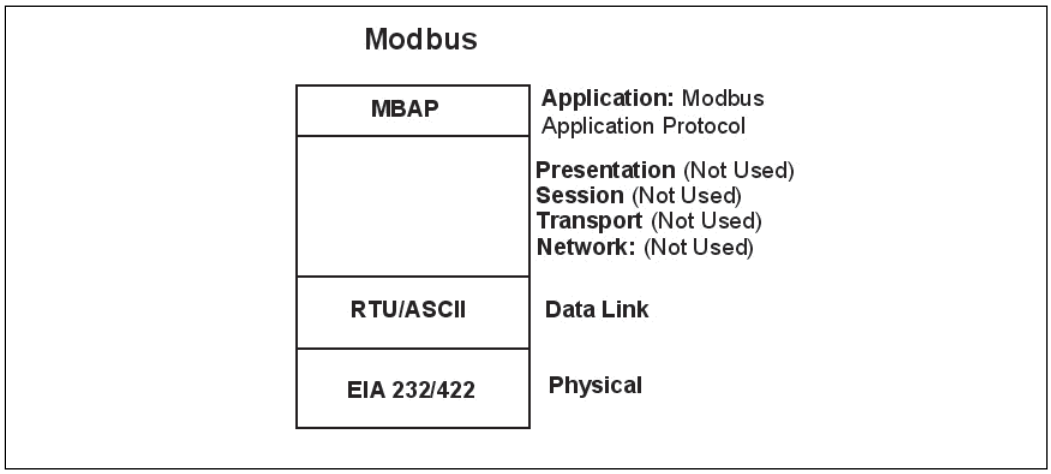


Figure 6-4. Modbus RTU/ASCII

The RTU mode has higher information density and is more widely used. There will be 246 addresses (1-247) available. Address 0 is an all-stations (broadcast) address. Speed depends on the devices attached, but typically goes up to 19.2 Kbps and covers distances up to four thousand feet. However, higher data rates are achievable at shorter distances.

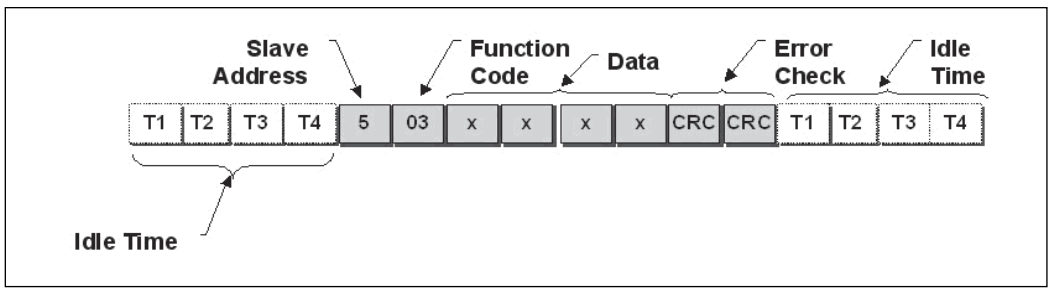


Figure 6-5. RTU Frame

As figure 6-5 illustrates, the ModBus Application Protocol (MBAP) assembles the slave address through the Data bits and hands it to the Data Link layer, which computes and tacks on the CRCC. The MAC and Physical layer use idle time as start and stop delimiters, using greater than 3.5 bit times to indicate that it is time for a new message. (Remember, only the master initiates these messages; this timing merely indicates to the slave when a message is to start or stop.) In fact, if transmission is dropped for longer than 1.5 bit times the message will be received as a fragment, and an error will be generated. Modbus also has an ASCII scheme, but it is not generally used at present.

Though the Modbus RTU method is more limited than the manufacturer's proprietary code, it will provide essential communications, and many integrators use it as their primary mode of communications between PLCs and host computers (PCs or otherwise).

Note that both PLC schemes, DataHighway and Modbus RTU, use standards-based Physical layers and are different in the Data Link layer (Media Access and Logical Link Control). Most PLC manufacturers also offer Ethernet (or IEEE 802.3) as a standard method of communicating from the master processor to other processors, workstations, operator stations, and gateways. Though this does away with incompatibilities between both Physical and Media Access Command (MAC) layers, a method must still be in place to address the various modules on the PLC, registers, and I/O. Again, the DataHighway Plus and Modbus RTU will usually be the data formats employed.

Selected Industrial Networks

In this section, the following networks will be explained in some detail, some more than others:

a. HART
b. DeviceNet
c. ControlNet
d. Ethernet/IP
e. LonWorks
f. AS-1
g. P-Net
h. Profibus/Profinet
i. Foundation Fieldbus
j. Ethernet and TCP/IP

HART

Highway Addressable Remote Transducer (HART), originally designed by Rosemount for their smart transmitter, has gained popularity and applications, ranging from transmitters to any entity in the two-wire loop.

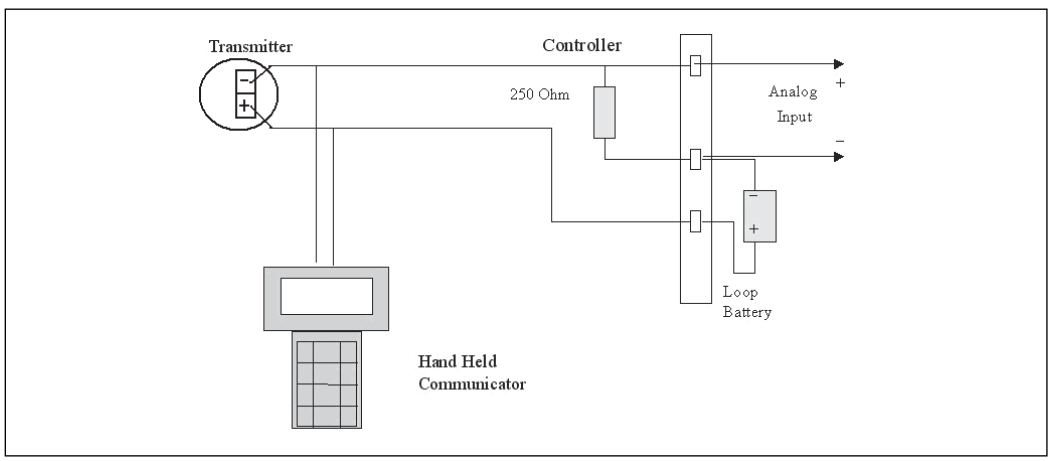


Figure 6-6. Hart Instrument with Handheld Configurator

Figure 6-6 is a block diagram of a two-wire loop that has a handheld communicator connected to it. Technically, the connection location is not important as long as a 250-ohm resistance (minimum) exists in the loop between the communicator and the power source. Basically, the HART protocol uses the Bell 202 Modem frequencies and handshakes. These provide a 1200-bps data rate over the shielded twisted-pair cable that is used in two-wire loop connections. Because the HART protocol is half duplex, most work (configuration or lengthy changes) is performed off line and then uploaded into the field device. The 4-to-20-mA signal is varied ±0.5 mA at either 1200 Hz (one) or 2200 Hz (zero). Just signaling on the loop will average out to a net zero DC signal, and the process will not be bumped. Most process controllers will limit the input rate of change to less than a 3 Hz change so signaling can go on without affecting the process. Changing ranges or outputs will bump the process, so they are performed with the loop in manual.

HART can be communicated with by using the handheld or a computer that has the appropriate modem card. Rosemount licensed the HART protocol to other vendors and eventually assigned intellectual property rights to the HART Communications Foundation, so it is an "open proprietary" code. As this is written, more than two hundred vendors are using the HART protocol to communicate not just with smart transmitters, but with valves, chart recorders—any device in a two-wire loop.

Of course, the number of devices that can communicate on one two-wire loop, be it the measurement or the control loop, is limited. HART-II, the multi-drop HART, can address up to six devices. HART-II is backward compatible with HART-I in that HART-I devices use address zero. Up to five other devices may be in the same loop provided each has a separate address. It is important to understand that even though you may address up to six devices, only one value will be represented by the 4-20 mA signal. Multivariable transmitters illustrate this limitation. Though these transmitters may measure process temperature, absolute pressure, and differential pressure, and use the intelligence to compute compensated flow, the 4-20 mA signal can only represent one of the values. A remedy for this is the HART splitter. Typically mounted in the control room, it communicates digitally with the transmitter to obtain each variable and produces (with supplied loop battery) a 4-20 mA signal for each variable. It should also be understood that only slowly changing process signals can be used with the splitter if control is desired. This is because 1,200 bps has a bit time of 0.83 milliseconds, and a whole HART frame may take nearly fifty octet times based on frame size and running half duplex, and you are looking at a considerable elapsed time between measurements. The actual time (which is for the maximum possible bit count) is certainly shorter than that stated at 0.83 milliseconds/bit but the caution still stands regarding real-time control based on the digital signals.

Though the analog 4-to-20 mA signal still appears in modern multi-drop configurations, it supplies the device power, but all variables are determined digitally and the 4-20 mA signal itself will represent one value that is converted from digital to analog in the transmitter. The

protocol is master/slave with up to two masters allowed (handheld and computer-driven interface). However, the masters cannot intercommunicate. Figure 6-7 illustrates a HART frame.

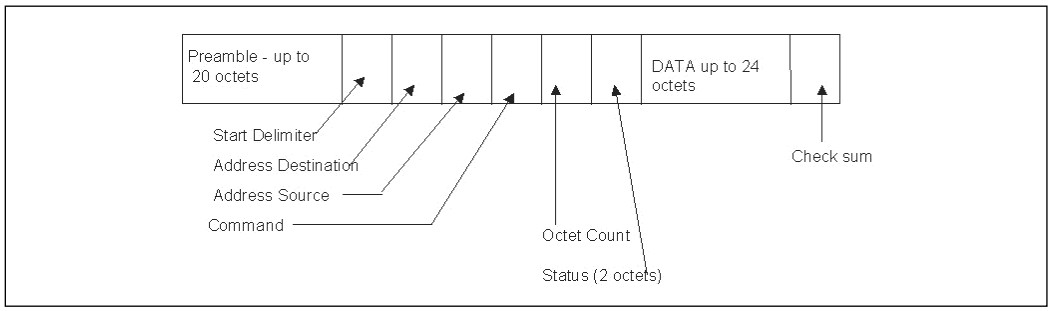


Figure 6-7. HART Protocol Frame

The preamble in figure 6-7 is a variable length of an FF (all ones) octet. Up to twenty may be sent. However, doing so could have deleterious effects on throughput, so each device will inform the communicator of its minimum required preamble. It is possible to send the probable number of preambles upon the first transmission (when the minimum for the device is unknown), the number of preamble octets most modern devices require is significantly less than 20. If a response is obtained, throughput has been increased. If not, then the maximum of twenty can be sent. Character parity as well as a check sum is derived at the transmitter, the same derivation is performed at the receiver, and if the checksums match, the frame is accepted. This process provides adequate error detection for the speed and the media.

Though HART is primarily used to communicate with smart instruments in the otherwise dumb two-wire loop, it is considered too slow for closed-loop control with many control and measurement points and is not normally used for control. Although nothing prohibits it from being used (as fieldbus is) as a supervisory system after all configuration has been accomplished, such a system would include all of the new HART transmitters, with PID function blocks included in the transmitter. This overcomes transmitting the process value and determining control constants at some other location. By doing PID locally at the transmitter only the set point need be digitally transmitted.

Smart positioners with HART communications circuitry are used on valves. They not only locate the valve stem position accurately and keep track of the number of strokes; some models can even transmit the inlet and outlet pressures as well as the process fluids temperature. This amount of information (as well as that provided by the transmitters) has never been available (outside of a small pilot plant or laboratory) and should greatly enhance process optimization, safety, and maintenance.

There is today a large installed base of HART instruments with ever increasing amounts of configurability and processing power. This type of network will therefore be around for

some time, or at least until the demise of the two-wire loop (the author offers no prophecies on that date). Adding HART devices to a two-wire loop means that only the devices and an interface have to be added. In fact, many DCS vendors will communicate with HART instruments directly over the two-wire loop. Doing so offers the benefit of exploiting the wiring that is there and represents an incremental rather than radical change. There was a time when people wondered what the benefits were of a smart instrument, as if increased reliability, self-diagnostics, greater accuracy and precision, and a cost similar to or less than conventional instruments weren't reason enough by themselves.

DeviceNet

DeviceNet is a low-level network that provides connections between simple industrial devices (sensors and actuators) and higher-level devices such as PLC controllers and computers. It provides master/slave and peer-to-peer capabilities, using the producer-consumer model (which allows devices to share information on a cyclical or event-driven basis). DeviceNet provides interoperability through open/sealed-type device connectors, diagnostic indicators, and device profiles. The DeviceNet network runs on three different cable types, referred to as "thick," "thin," and "flat." The thick and thin are round cables and vary in the amount of current they can carry, while the flat cable is unshielded and has one pair for power, one pair for data, and a mechanical key to ensure proper connection.

DeviceNet is based on the CAN (Controller Area Network), which has found significant usage in the automotive industry linking automotive computers together. (Figure 6-8 illustrates DeviceNet installation concepts.) DeviceNet uses a modified CSMA system with arbitration, so there are no collisions. CAN chips are inexpensive, in the $1 to $2 range (2007).

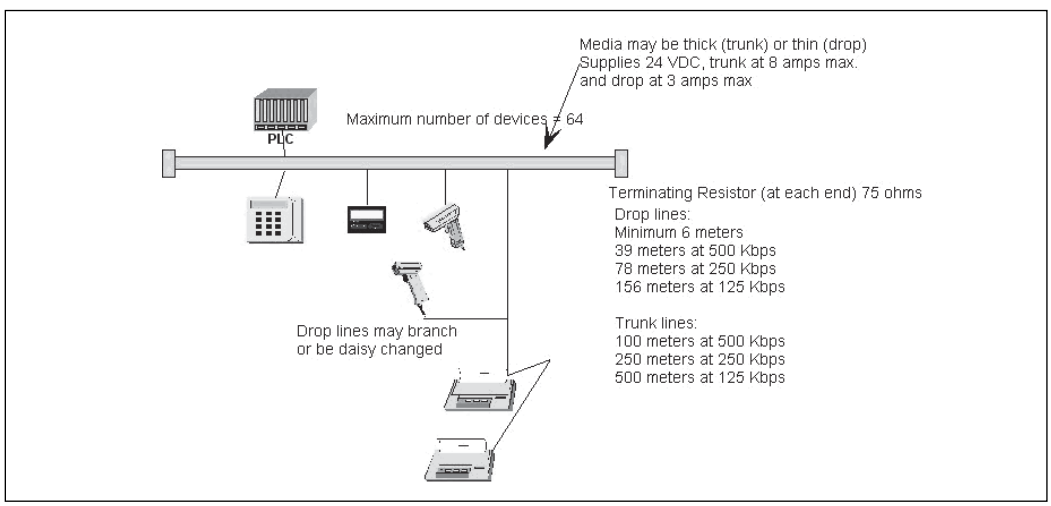


Figure 6-8. DeviceNet Topology

Arbitration of the DeviceNet bus prevents collisions from occurring and guarantees that one of the messages will be completed. The Identifier field contains a unique pattern of bits that

at configuration time will determine the priority of the device. The lower the identifier, the higher its priority. Zero states have precedence over 1 states. So if, after determining that the line is idle (just like Ethernet), two devices both attempt to transmit at the same time, the node that is transmitting a 1 from its identifier stops transmitting if it hears a zero and the other node continues. This is a bit-wise arbitration, and the one with the first zero has the bus.

ControlNet

ControlNet is an open network that supports simultaneous I/O and explicit messaging. It implements the producer-consumer model of communications and supports multiple masters, peer-to-peer and broadcast communications. ControlNet operates at 5 Mbps, is deterministic, and is intended for controller-to-device, controller-to-controller, and controller-to-external systems that provide seamless integration between DeviceNet and Ethernet. With the correct module, ControlNet interfaces quite well with Foundation Fieldbus. It is approved for installation in intrinsically safe (IS) environments. Figure 6-9 shows a conceptual diagram of ControlNet.

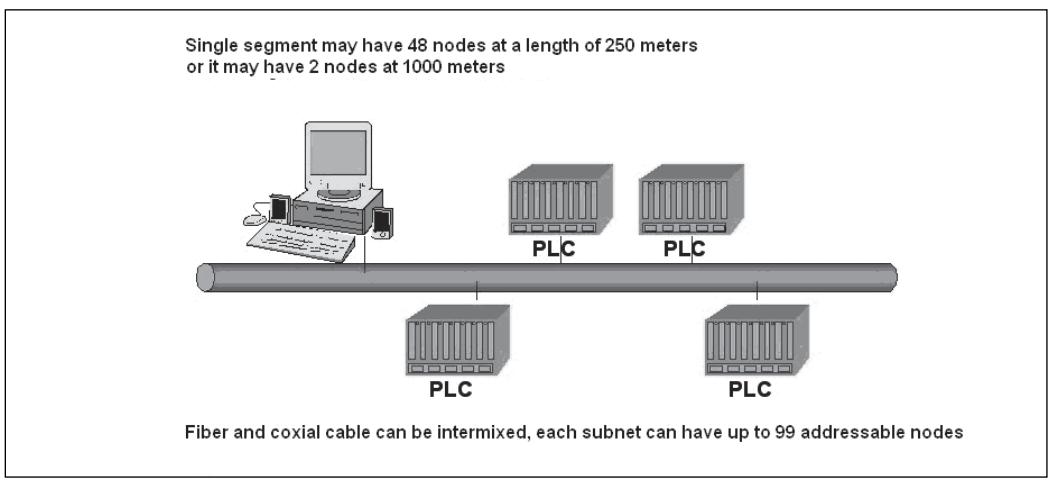


Figure 6-9. ControlNet Topology

ControlNet was developed by Allen-Bradley, but since July 1997 it has been controlled by the ControlNet International organization of vendors and users.

Ethernet/IP

By piggybacking on established Layer 1, 2, 3, and 4 protocols (notably Ethernet and TCP/IP), Ethernet/IP has standardized its first four layers. The "IP" in Ethernet/IP stands for Industrial Protocol and not the Internet Protocol. Figure 6-10 illustrates the relationship between OSI and Ethernet/IP.

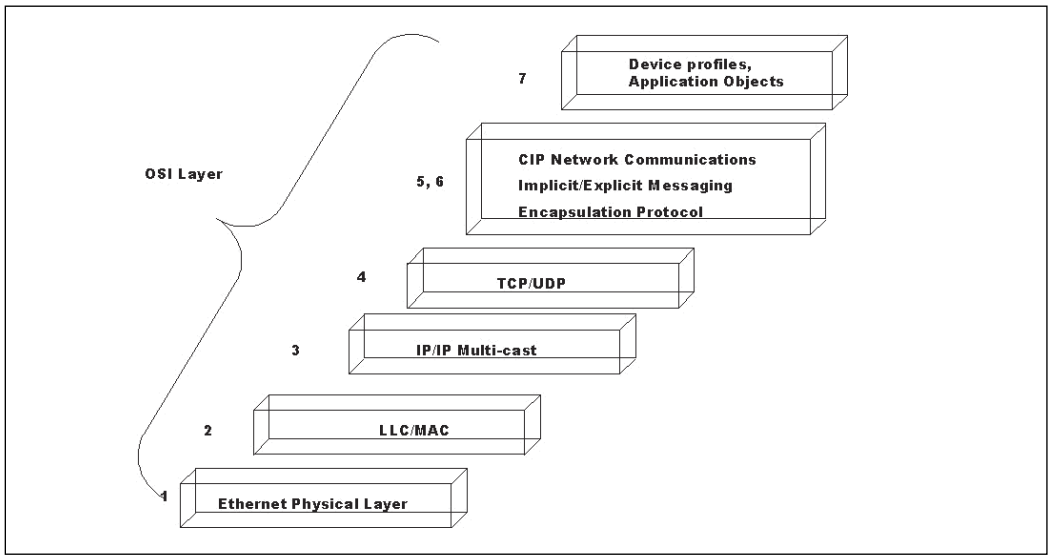


Figure 6-10. OSI and Ethernet/IP

What Ethernet/IP offers is the same application layer used in DeviceNet and ControlNet (Layer 7—along with 5 and 6) and the Control and Information Protocol (CIP). Frame size is 1,500 octets (Ethernet Standard), and the frame format is different from DeviceNet/ControlNet in that their framing is encapsulated in the TCP/IP frame. One of the main differences between ControlNet and Ethernet/IP is that ControlNet's Schedule Segment and Schedule Object are not used in Ethernet/IP. Table 6-1 compares some of the three systems' salient features.

Feature	DeviceNet	ControlNet	Ethernet/IP
Topology	Passive trunk line	Passive trunk line	Active Star
Determinism	High	High	High (full duplex switched)
Layer 7	CIP	CIP/Routing	CIP Routing
Standards Body	ODVA	ControlNet Int.	IEC 61158
Data rate	125-500 Kbps	5 Mbps	10/100 Mbps
Max Seg. Length	500 meters	1 Km	100 meters
Max Network Length	4 Km with repeaters	30 Km	Unlimited
Maximum Nodes	64	99	Unlimited
Intrinsic Safety	No	Yes	No

Table 6-1. Comparative Features Table

The encapsulation is a normal part of TCP/IP, which regards the CIP protocol as data. There is a 24-bit header after the TCP header. However, in the TCP header the target port is set up as 44818 (decimal or AF12 Hex).

LonWorks

LonWorks is based on the EIA 709 standard. It uses a differential Manchester signaling system and a modified CSMA media access called "predictive p-Persistent," which improves

performance in heavily loaded buses. LonWorks uses a special IC—the Neuron Chip—and is a full implementation of the seven-layer OSI model. LonWorks uses the domain model (hierarchical) for addressing and four major message types: acknowledged, request/response, unacknowledged, and unacknowledged repeated. Rather than file oriented, the LonWorks protocol is message oriented, using short sensor and control messages. Its data rate approaches five hundred messages a second, which is approximately 1.25 Mbps. A complete LonWorks network can have a maximum of 248 domains. Within each domain there can be 255 subnets, and each subnet can have 127 nodes. Variables are defined for each node (up to 255). LonWorks has a very credible performance according to the manufacturer, particularly in building automation, its major market. It might be a formidable competitor for other fieldbus products, except that it is not commonly used for process automation outside of building controls. Figure 6-11 illustrates the relationship between LonWorks and the OSI model.

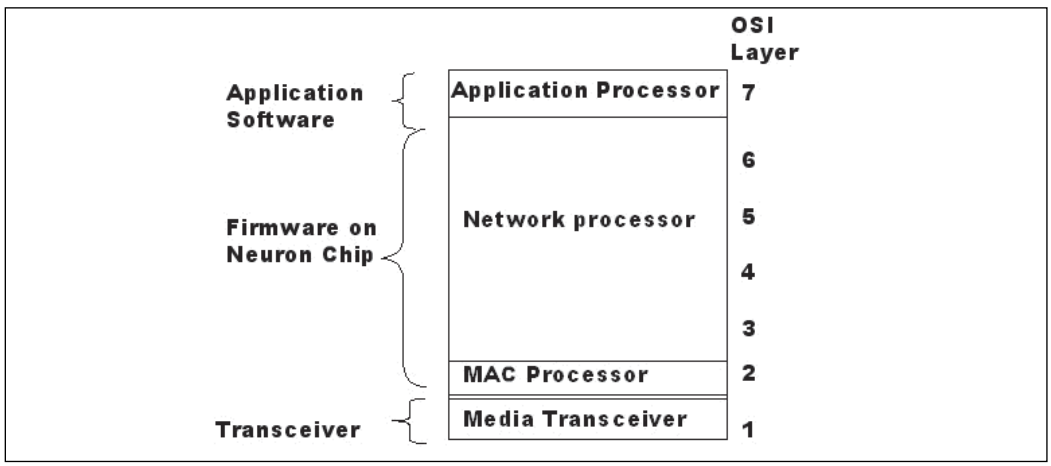


Figure 6-11. LonWorks and OSI Model

The p-Persistent CSMA uses a randomized slot selection and a priority set of slots that is determined at installation time. This is actually a collision-avoidance scheme rather than a collision-detection scheme. No collisions are possible during the priority time slots, and a predictable round-trip time can be made for high-priority messages. A collision-detection scheme is optionally available for lower-priority messages. The packet format in figure 6-12 illustrates how the data is moved from address to another address. There are addresses for unicast (one-to-one) and multicast (one-to-many). The messages may be acknowledged or unacknowledged, with the latter conserving bandwidth.

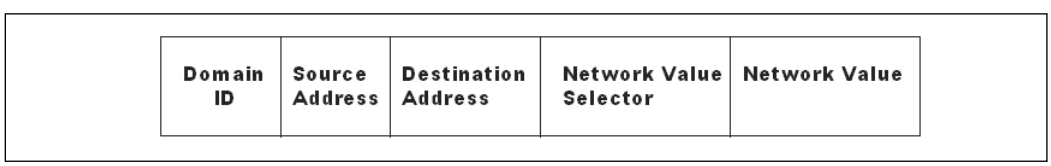


Figure 6-12. Lon Packet Format

LonWorks has found acceptance as a sensor network, a device network, and a field network. LonWorks Networks Services (LNS) supports interoperable tools, Windows (95/NT/2000) HMI, Component Software (ActiveX) objects, and local, remote, and Internet access. The EIA 852 standard covers LonWorks on Ethernet. Echelon, the manufacturer of the Neuron chip used in LonWorks, has recently made their source code available so other manufacturers can implement the source code without the need to embed a Neuron chip.

AS-i

The Actuator Sensor Interface (AS-i) was developed for the European market in 1993 and brought to the United States in 1996. It is a low-level technology that complements rather than replaces PLCs and fieldbuses. AS-i consists primarily of an untwisted, unshielded two-wire cable that connects devices with the AS-i chip. The chip has no processor and therefore needs no programming. It handles bit data only and is fast, secure, and built for the industrial environment. Figure 6-13a illustrates how AS-i fits into the hierarchy of buses.

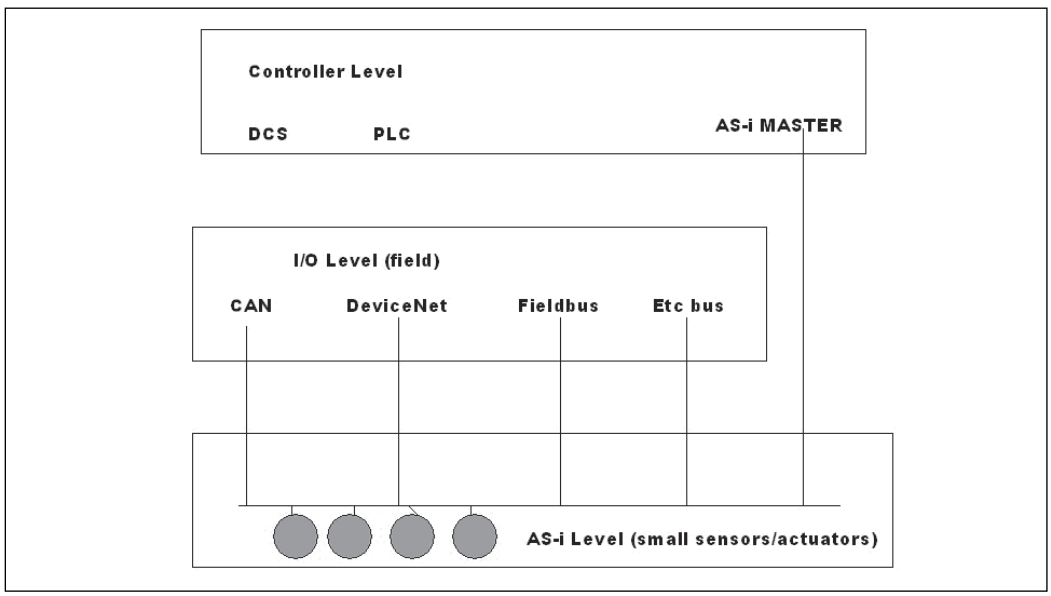


Figure 6-13a. AS-i in the Hierarchy of Buses

In AS-I, the master can be the controller, or it can be controlled by being coupled into a fieldbus. The two-wire cable carries the data and power and connects using a "piercing" technology, which makes for easy-to-install connectors. AS-i has up to IP67 protection. The AS-i protocol is very simple: master/slave. It is performed on a ring or cyclical basis, where a poll is sent one at a time to each registered slave. The published time for complete rotation with thirty-one slaves (the maximum you may have is thirty-one slaves and one master) is five milliseconds. It should be noted that even the AS-i master operates under its own firmware, and no programming is required of the user. With only thirty-one slaves, it is still possible to achieve 248 binary signals per network. The maximum length without repeaters

is one hundred meters, and you can use two repeaters to get the maximum length of three hundred meters. The AS-i may have almost any topology you wish to create: star, daisy chain, tree, or branch. This system can definitely reduce wiring and installation costs, not to mention headaches.

All AS-i connections use coupling modules at their base (see figure 6-13b). On top of the couplers are placed one of the following:

- A tap cover
- A passive distributor
- An application module

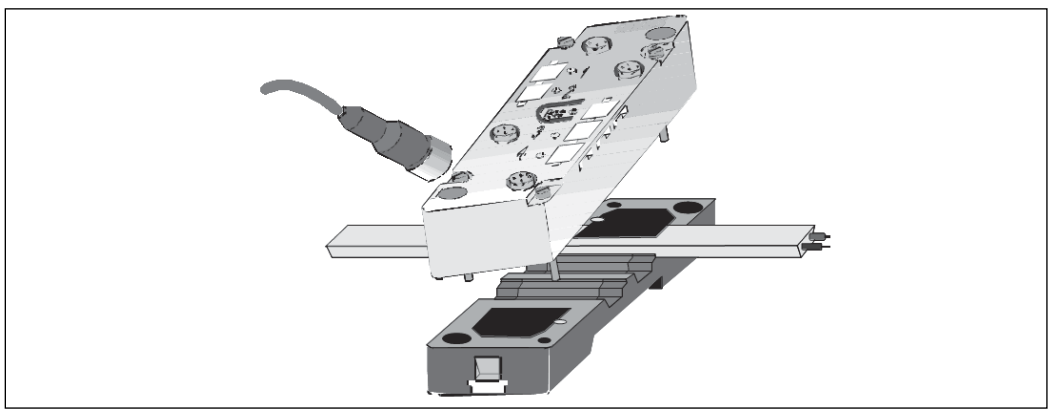


Figure 6-13b. AS-i Connector

P-Net

P-Net is based on the EIA 485 standard. It has an allowable cable length of 1,200 meters without repeaters. Its topology is illustrated in figure 6-14. P-Net signals at a 78,600 bps rate. Using a technique called "parallel" (most refer to it as duplex), it will receive the acknowledgment from the received device as it is completing the transmission of the frame to that device. This "pipelining" speeds up data transfer, and the P-Net people claim it is as fast as any 500 Kbps system.

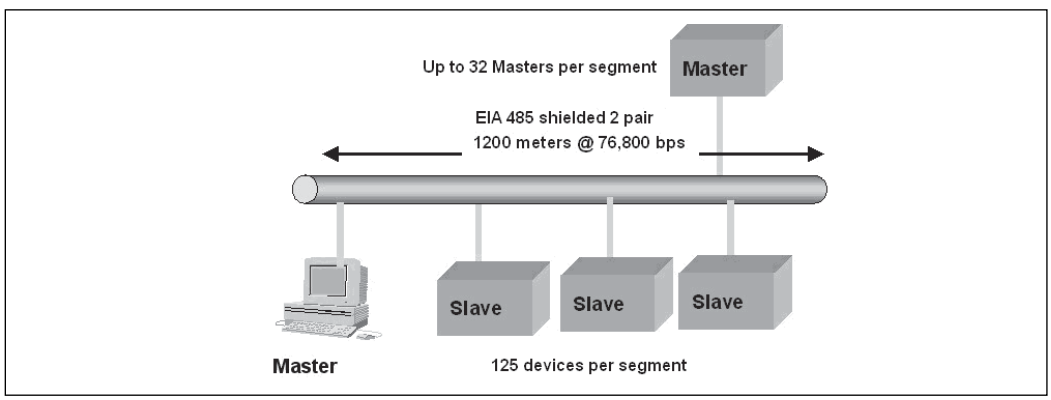


Figure 6-14. P-Net Topology

P-Net's marketing people say it uses a "virtual token" when actually it is a counter-controlled bus mastering technique. It accords 10 bits to each master after the minimum of 40 bus idle bit times are counted. As an example, assume Master 3 (not shown in figure 6-14) had been transmitting and receiving. After the bus has been idle for 40-bit times, the idle bus bit-period counter is incremented one, and Master 4 (not shown) may have the bus. If Master 4 has nothing to say (or isn't even there) the counter keeps counting until fifty is reached, plus one. At this point, Master 1 (the computer) could have the bus. If Master 1 has nothing to say then when the counter hits sixty plus one the second Master (the box labeled Master in figure 6-14) could transmit. If Master 2 is so inclined, the idle counter will be reset to zero until Master 4 has completed its transmission. Since no data is transferred over the bus, this process is deemed an efficient way to handle "token passing." P-Net is object oriented and uses OLE2 (Object Linking and Embedding) as a way for the Application layer to access physical objects via a "virtual object." P-Net has Visual Basic for Applications as a programming structure.

Profibus/ProfiNet

Profibus is actually three different protocols. Profibus FMS (Fieldbus Messaging Specification) is a general-purpose solution for peer-to-peer communications tasks. Profibus DP is a high-speed data communications network for factory automation; and Profibus PA is for the process automation market. The Physical layer has three transmission technologies available: EIA 485 two-wire copper cable, fiber optic when electromagnetic compatibility (EMC) protection is required, and IEC 61158-2, a two-wire copper cable with provision for providing power over the bus. The Physical layer options are outlined in figure 6-15.

EIA 485	Fiber-Optic	IEC 61158-2
Asynchronous 9.6 Kbps to 12 Mbps	Asynchronous 9.6 Kbps to 12 Mbps	Synchronous 31.25 Kbps
Shielded Twisted Pair	Mono-mode, Multi-mode Plastic, PCS/HCS fibers	Shielded Twisted Pair IS and bus power options
32 stations per segment 126 stations maximum		10-32 per segment (depending upon power) maximum of 127 stations
Distance: 12 Mbps 100 m 1.5 200 m 187.5 Kps (or less) 1 Km		Up to 1.9 Km depending upon power requirements
Repeaters Allowed	Extendable to 100 Km	No Repeaters

Figure 6-15. Profibus Features

Profibus differentiates between master and slave devices for access and uses a technique similar to a floating bus master. The slaves are always polled, whether there is just one master or multiple masters. The only passing of command is performed between the masters. Though the physical layers differ, Layer 2 is the same across all three Profibus implementations. FMS and DP use the same cable, and their signals can be combined.

Profibus PA is dedicated to the process industry and uses a different physical media technology then DP/FMS. Profibus will support bus, tree, or star topologies; the tree is the preferred one.

The Profibus Data Link layer uses virtual field devices (VFD). It was intended to replace the EPA MAP (Enhanced Performance Architecture—Manufacturing Automation Protocol), using the Manufacturing Messaging Service as an application utility. Unfortunately, MAP kind of went off the edge of the world, and though the MMS was evaluated by the fieldbus (SP50) committees, they ended up defining the User layer instead. Profibus is a German standard (DIN 19245) and an IEC 61158 standard. It has been successful in the discrete manufacturing segment, so much so that the Profibus Trade Organization states they are the only complete and proven solution for both manufacturing and process. This statement, of course, must stand the test of time, and not a few other vendors would tend to disagree with it.

ProfiNet is a continuation of the Profibus system. It is an object-oriented approach that uses standardized interfaces and technologies such as TCP/IP and COM/DCOM. ProfiNet is an attempt to unify the hierarchy from the sensor to the enterprise network. By using market standards such as XML, COM and DCOM over Ethernet TCP/IP, and OPC it provides a set of open, transparent, and integrated communications protocols. ProfiNet is an attempt to move Profibus from a distributed I/O system to a distributed intelligence system, where the intelligence is in the field devices rather than centralized in a control facility or cabinet.

Foundation Fieldbus

Foundation™ Fieldbus is designed to interconnect field instruments using a distributed intelligence system that puts the controller back in the field. As such, it must have low node costs, offer extremely reliable operation, be relatively simple to operate, and above all be a "real-time" system in which you acquire data and still have time to operate on it. Foundation Fieldbus handles data in two different modes depending on whether the data is cyclic (operational data: traffic and algorithms) or acyclic (background data: configuration and diagnostics). Cyclic traffic is generally low volume and time critical, while acyclic is just the opposite: high volume and not time critical.

Figure 6-16 illustrates the fieldbus's OSI model. Note that it uses Layers 1, 2, and 7 of the OSI model but in actuality has four layers. The fourth layer is the User layer, which is defined in the specification. Stating that it has four layers is not totally accurate, however. Since Foundation Fieldbus is segment routable, the Applications layer handles the majority of the Layer 3 functions for the H1 fieldbus.

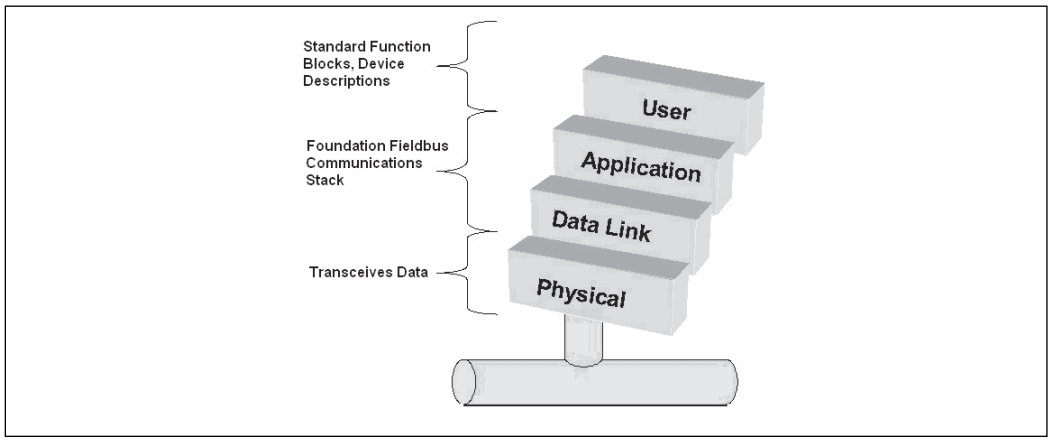


Figure 6-16. Fieldbus Model

The Physical Layer

The Physical layer connects the node to the media and provides for activation, deactivation, and maintenance on a bit basis (octets or byte patterns are not recognized). The SP50 standard as implemented by the Fieldbus Foundation defines types of media, signals, speeds, and topology, including the number of nodes and node power.

Continuous process control requires a moderate speed, and speed requirements dictate power consumption. The higher the speed, the more power is required. Large power demands are inconsistent with the intrinsic safety requirements typically found in continuous processing. The designers of Foundation Fieldbus decided on a moderately high-speed bus for intrinsic and field instrument work (H1) and originally selected two higher speeds for nonintrinsic areas (H2). Each had its own advantages. It should be noted that all devices on one bus had to have the same speed and options for the media. However, powered and nonpowered could be mixed, and on H1 intrinsically safe and non-IS devices may be interconnected.

The first physical media characterized in the Foundation Fieldbus standard is copper. Standards are currently being developed for fiber-optic media. The H1 standard transmission rate is 31.25 Kbps (the same as Profibus PA). It takes a long time to generate a standard and then populate it with actual equipment. Wisely, the Fieldbus Foundation abandoned the effort to develop a 1- and 2.5-Mbps H2 and embraced instead a proven, workable, cost-effective commercial standard: HSE, or High Speed Ethernet that is duplex switched at 100 Mbps therefore is both deterministic and fast. HSE connects fieldbus segments together, connects a fieldbus to other networks, and ties operator consoles together.

H1 transmission (31.25 Kbps) is the synchronous, half-duplex serial signal that uses Manchester encoding and is therefore self-clocking. Manchester encoding places the transmitters' output state into transitions (rather than plateaus), and timing can then be

recovered at the receive device(s). Preamble and start and stop delimiters are not Manchester encoded and therefore are instantly recognized as such.

Signal

With current modulation, the devices can sense the voltage drop across the terminating resistors (100 ohms each). For H1, modulated current (Manchester encoded) is 15 to 20 mA peak to peak, and the typical sensitivity of the receiver is 150 mV.

Topology

In fieldbus, bus or tree topology is supported by the 31.25 Kbps scheme. A bus has a trunk cable with two terminators. Spurs attach to the bus by way of a coupler. A spur may contain more than one device, and it may have an active coupler to extend its length. The trunk (bus) may have an active repeater to extend its length.

An installation guide follows:

1. One fieldbus segment can have between two and thirty-two (thirty if you allow for power supply and monitor) non-I.S., non-bus-powered devices. Alternatively, it can have between two and six powered devices for an I.S bus of which between one and four of the devices are in the hazardous area. Finally, it can also have between one and twelve bus-powered devices located at the end remote from the power supply. A system can have more than the listed devices because devices are calculated to draw 9 mA (±1 mA). So if the devices in fact draw less than that you can have more devices. The I.S. areas were calculated for a 19 VDC (output) barrier providing 40-60 mA. The total of twelve bus-powered devices is based on the assumption of a source of 20 VDC.
2. The total cable length cannot be more than 1,900 meters. This not the geographic distance but the total distance of all spurs and trunks.
3. There can be no more than four active couplers or repeaters.
4. Maximum propagation between any two devices on the segment cannot exceed twenty nominal bit times.
5. Devices can be connected or disconnected without interrupting operations of the bus, and any errors caused by connection or disconnection must be detected and corrected.
6. The failure of any communications element (except for such failures as jabber, short circuits, or low impedance) cannot interfere with other transactions for more than 1 ms.
7. Connectors will be uniquely marked with physically guided connectors in order to maintain correct polarization.
8. Attenuation factors for different topological arrangements must be determined to ensure that a particular topology will meet the signal power budget
9. For redundant media, each cable has to meet all the network rules, and there can be no non-redundant segment between two redundant segments. Repeaters will be redundant, and if they are transmitting on more than one channel, there can be no more than five bit times' difference in propagation time for any two devices on any two channels.

Figure 6-17a illustrates connections in an H1/H2 network. The high-speed backbone is at 100 Mbps and can connect to standard Ethernet equipment. This architecture allows for the use of standard low-cost wire and fiber-optic media with fault-tolerant communications and linking devices.

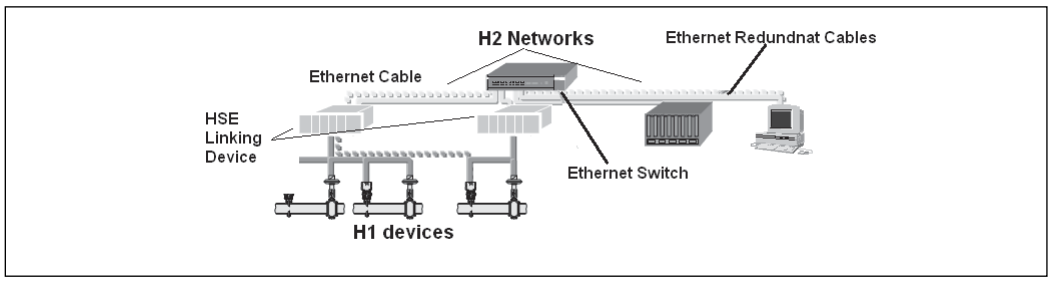


Figure 6-17a. Fieldbus H1/H2 Connections

Figure 6-17b illustrates the salient requirements for a repeated network.

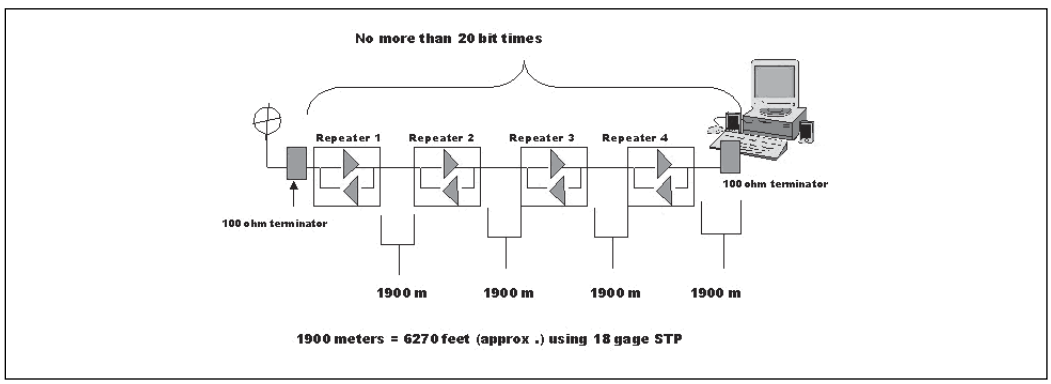


Figure 6-17b. Fieldbus Repeaters

Additionally, each node will have a hardware antijabber or jabber-inhibiting self-interrupt capability, which allows transmission from the jabbering node to last no longer than between 120 to 240 ms.

Data Link Layer

As is, IEEE 802.X, the fieldbus Data Link layer, is divided into two (actually three—one is an intermediate layer) sublayers: Fieldbus Media Access Control (lower) and Fieldbus Data Link Control (upper).

For access control, fieldbus modifies the token-passing bus arrangement of rotating the token and instead opts to use a method similar to the DataHighway Plus floating bus master. Here the master is called an *Active Link Scheduler* (ALS) and uses two different tokens, specifically the *delegate* token and the *reply* token. Figure 6-18 illustrates the delegate token.

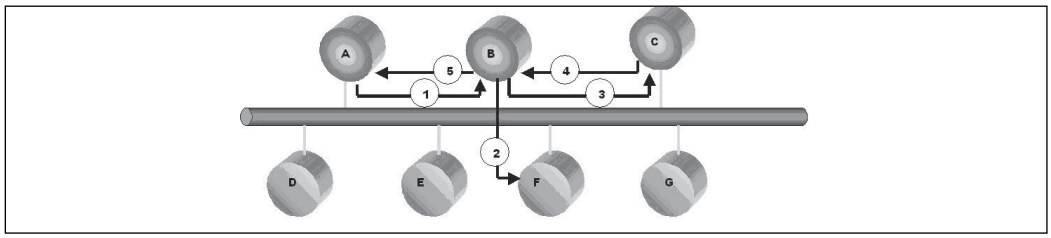


Figure 6-18. Delegate Token

In figure 6-18, the designated Active Link Scheduler (Station A) passes the token to Station B. As long as the token is in the delegated station's possession (a prescribed time that depends on network loading, configuration, etc.), the station may transmit and request replies from other stations. At the prescribed time, it must return the token to the ALS, which then apportions it out to the next station. Figure 6-19 illustrates the reply token. In figure 6-19 the ALS (Station A) passes the token to Station B. Station B may have one initiated transmission (to one or more stations) and then must return the token to the ALS.

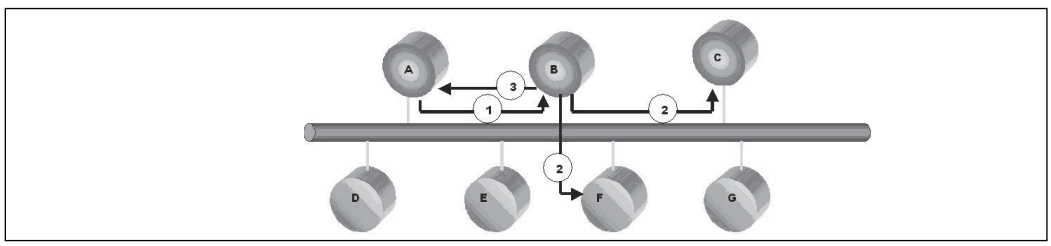


Figure 6-19. Reply Token

Fieldbus Layer 2 Frame

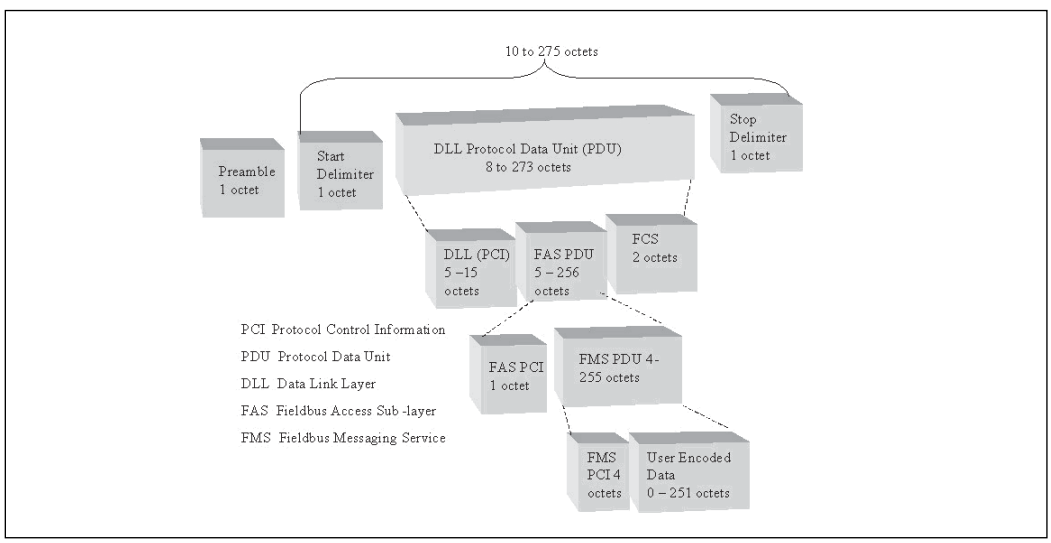


Figure 6-20. Fieldbus Frame

Figure 6-20 illustrates a fieldbus frame. Note that it has a preamble, start delimiter, destination DLL PCI, data block, frame check sequence (CRCC), and a stop delimiter—the same format that has been observed in almost all protocols since LAP-B. The data (all bits except those in the preamble and delimiters are called data bits not just those in the data block) is placed on the wire using Manchester encoding. The data is a valid Manchester-coded signal; the preamble and start/stop delimiters are actually "nonvalid" (not data) signals and are easily discerned from the data signals.

Application and User Layer

The higher layers are implemented in object-oriented design (OOD), in which objects are used rather than lines of linear code. This greatly increases programming productivity, reduces the level of abstraction required, and allows the product to be visually oriented, an essential benefit for users given fieldbus's intrinsic complexity.

In fieldbus, background processes (such as configuration, messages, etc.) are acyclic messages and use a form of the client/server model. Recall that a server is any node that shares its resources, and a client is any node that uses a server's resources. For the cyclic processes (those that are time dependent, such as a PID algorithm), fieldbus uses a publisher-subscriber model. This method is derived from the more frequently used producer-consumer model. The main difference between the two models is that publisher-subscriber normally requires the subscriber to subscribe to a certain event controlled or reported by the publisher and then periodically requests that the publisher publish this data, and those that subscribe to that publisher utilize the data This is called a "push" process. In producer-consumer models, typically the producer puts the data on the network for all (broadcast), and those that require it consume it, this is also a "push" operation. When a device requests information from a server then it is usually called a "pull" process.

Fieldbus builds applications using function blocks. From the User layer perspective, any device is more than just parts; it is a parameterized network node. So the User layer sees the nodes as virtual field devices (VFD), which are the interface between the communications protocol and the function block. A node may have one or more VFDs.

Both system and network management rely on the tightly coupled Applications/User layer. System management is concerned with the following five issues:

Device Tag Assignment: Before a device is placed on the network it must have a physical device tag assigned to it. This is done using standard facility guidelines for the tag.

Station Address Assignment: Station addresses are assigned by a "plug-and-play" method in that a new device has a default address, and (assuming that it does not have a tag identical to some other device's in the network—which

shouldn't happen if facility guidelines are followed), the network assigns an address to the device.

Clock Synchronization: Clocks are synchronized from a master clock, so all devices maintain the same real time.

Scheduling of Application Processes: Scheduling is required to ensure there is neither dead time while waiting for execution nor variance in timing delays. The priority is as follows: (1) function block (like PID or AI) execution that is scheduled in the field device, (2) communications (operational traffic) that is scheduled in the ALS (master), and (3) background (acyclic) communications.

Function Block Binding: Function blocks are distributed throughout the network; there is no centralized location for control. In effect, the network is the controller. Communications links the function blocks. When a new block is entered into a system it must be bound. That is, its stack and the other devices that are interconnected with it must have the service and destination access point addresses (actually, offsets from the block address) available and stored throughout the system devices that will use the information. In other words, once a block is instantiated, it then goes through a binding process that the software uses to identify function blocks, determine methods and properties, and connect the device.

Figure 6-21 illustrates the concepts of the VFD and function block in terms of the OSI layers for the communications stack.

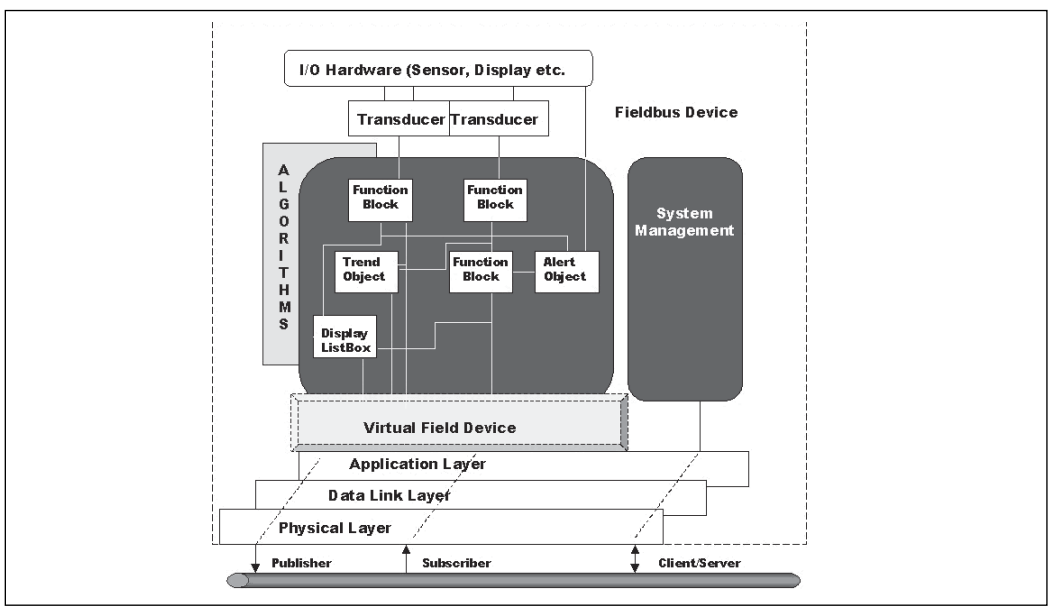


Figure 6-21. Function Block

Function blocks contain an algorithm and a set of parameters (parameter block) for processing inputs and producing outputs. The function block is an object—an abstraction of data and software. It exists only in the software model but is useful in enabling humans to organize and recognize. Each block has a unique tag (just like a physical control system), assigned by the user. The block tag and the parameter tag uniquely identify all parameters. A sample-and-hold system is used, in which an instantaneous picture is taken of all inputs. This keeps them from changing while they are being processed by the algorithm for that block. The outputs are then updated and published to the network.

To build a system, you essentially assign tags and select a control strategy by selecting function blocks, linking them, and setting the parameters. Initially, there were ten standard function blocks; but many have since been added.

The ten standard basic blocks are:

- Analog Input block (AI)
- Discrete Input (DI)
- PID Control block (PID)
- PD Control block (PD)
- Analog Output block (AO)
- Discrete Output (DO)
- Control Selector (CS)
- Ratio (RA)
- Bias (B)
- Manual Loader (ML)

In addition, some of the added blocks are as follows:

- Analog Alarm
- Discrete Alarm
- Splitter
- Dead Time
- Lead/Lag
- Pulse Input
- Calculation Block
- Set-point Generator
- Integrator
- Step Control
- Output Signal Selector
- Complex Analog Output
- Complex Digital Output
- Device Control
- Arithmetic
- Analog HMI
- Discrete HMI

As fieldbus advances, even more blocks will be added.

A pressure transmitter will likely contain an AI and a PID function block. An equivalent of the I/P (Fieldbus to Pneumatic) will contain an AO and a PID block. A large number of devices exist, and they contain a large number of function blocks. The control system and strategies employed are all in how you connect them. Figure 6-22 is a block diagram of a function block.

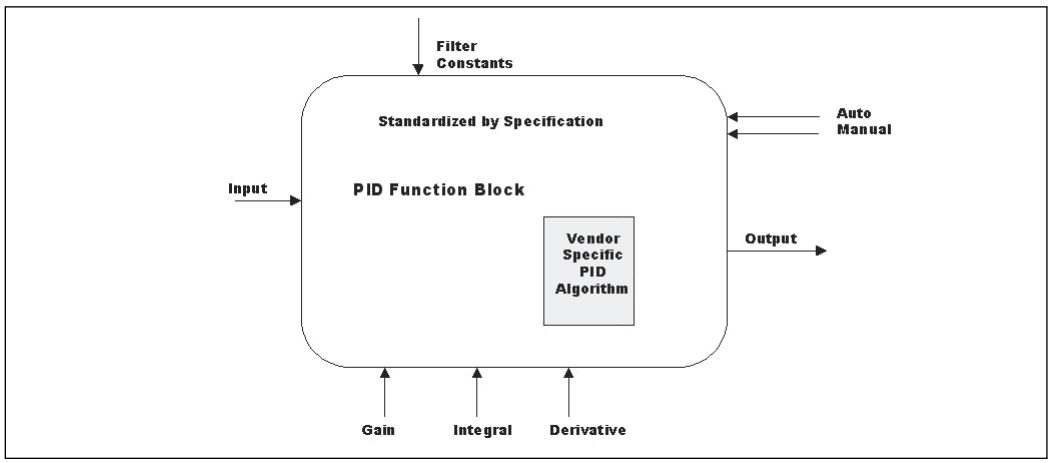


Figure 6-22. PID Function Block Diagram

The function block inputs and outputs are standardized for all vendors. However, it's up to the vendor how it produces the output from the inputs, and that is what distinguishes one vendor's product from another's. Note that this method of ensuring compatibility will make all devices (at least functionally) interoperable and interchangeable. Some, I am sure, will perform better than others under certain conditions in certain applications. This fact will allow vendors to develop a brand name that is associated with a certain level of performance and cost effectiveness.

Hierarchy of Buses

The H1 fieldbus is intended to connect to field devices, but the higher-speed mode, HSE, is intended for intersegment or internetwork connectivity. Figure 6-23 illustrates the hierarchy of buses.

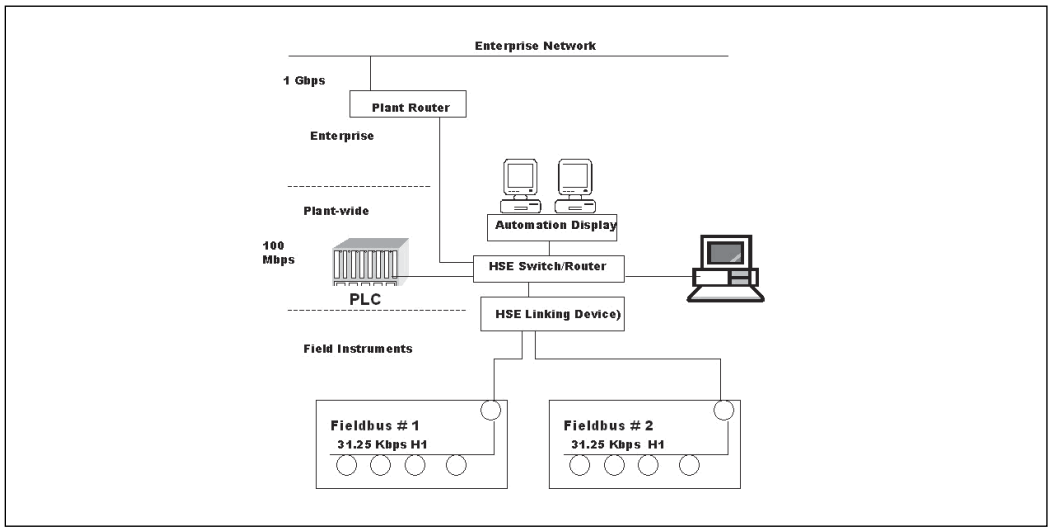


Figure 6-23. Hierarchy of Buses

The Fieldbus Foundation (a nonprofit vendor-supported organization) tests protocol stacks and devices to ensure that they conform with the specifications. If you purchase a device that has its seal of conformance, you can be sure it will interoperate and meet the specifications.

Fieldbus Summary

The advantages of fieldbus are as follows:

It is designed for process control,
It is a real-time system with node-synchronized clocks,
It has an open system standard,
It offers high information flow,
It provides better accuracy,
It offers higher reliability,
It provides good control-loop performance,
It is easier to calibrate and maintain,
It offers interoperability and interchangeability.

Ethernet/TCP

The commercial version of Ethernet was explained in chapter 4, on LANs. Here we will just explain why the author and many others believe it will soon be in an industrial network near you.

First and foremost, it is the least-expensive per-node per-bit-per-second network available today.

Second, it has been proved in installations of many different sizes and under many different conditions. One of the first arguments against the adoption of Ethernet in industrial settings (and it was the author's in the heydays of MAP/TOP) is that Ethernet is not deterministic:

there is no guarantee when a node can get on the network. If you are referring to a half-duplex, hubbed network running CSMA/CD, you are right about the guarantee. But that does not mean "never," nor that there is no method to achieve reliable (guaranteed) round trip times. Ethernet (of the shared media, hubbed variety) is based on probabilities. So is your phone system. Assuming that your line is intact to the central office and that your phone is working, when was the last time you did not receive a dial tone when you picked up the handset? Yet in a residential area there are only enough dial-tone generators for about 10 percent of the subscribers. Why? Because statistically the chance of more than 10 percent of the subscribers originating calls at the same time has an extremely low probability. And if they did, you would probably just hang up and pick the handset up again. Then you should have dial tone, since others will have hung up.

Another of the deterministic arguments is that you cannot get an accurate time slice at each time interval to properly calculate PID control. One of the many myths surrounding Ethernet is that if it experienced heavy loading (and the author has heard values of from 30 percent to 70 percent) it would bring the network down due to collisions and exponential time-outs. In their "Measured Capacity of an Ethernet: Myth and Reality," Boggs, Mogul, and Kent showed that in practice Ethernet delays are linear and range from 2 ms for a lightly loaded network to 30 ms for a heavily loaded network (and this was a 10 Mbps network).

Obviously, the loading of a measurement and control network is different than an office network, yet the solutions are the same. Don't heavily load the network. Just restrict the number of devices in each collision domain so traffic does not peak over 10 percent of capacity. The fact that this loading is not a real problem should be understood since most PLC vendors have had an Ethernet port for sometime. To truly make an Ethernet network deterministic you only have to employ full duplex switches. Because each node thinks it has full bandwidth (in fact, twice as much as a half-duplex network) and always has the bus, collisions do not occur, and there are no time-outs. This is the option specified for fieldbus in the HSE.

The use of Ethernet switches (rather than hubs) allows IEEE 802(q) and 802(p) standards to be implemented. Both of these standards are quality-of-service (QoS) rules, and the switch assigns priorities to different traffic. And then there is the solution that many proprietary systems use. They determine timing in Layer 7 by time-stamping, dispatching, or otherwise allowing Layer 7 to prioritize and decide communications (rather than first come, first serve). In fact, in their "Ethernet Rules Closed Loop System," Edison and Cole demonstrated for Hewlett-Packard a protocol on top of Ethernet that held timing uncertainties to 200 nanoseconds (enough for all but the most fastidious processes).

The problem with Ethernet, and indeed with employing TCP/IP above it, is that while you could "ping" everybody on the network (Ping is a low-level utility that ensures connectivity over a TCP/IP network), you may have a problem communicating unless there is a standard

Layer 7, which, at present, there is not. Another problem is commercial hardware that does not meet industrial specifications. The hardware that does is not "standard," although mounting most of the distribution in the control room alleviates the industrial requirements somewhat. Since the popularity of Ethernet in the industrial setting is rising, a number of hardware manufacturers have released "hardened" Ethernet products. These include industrial switches that use a ring for redundancy, water- and vibration-proof connectors, and many copper-to-fiber converters (to avoid unshielded twisted-pair cable on the plant floor). Indeed, just about any component of an Ethernet distribution network is available hardened—for a price, which is becoming more competitive by the week.

Since IEC 61158 approved the "standard" fieldbus (there are eight, including Profibus, Foundation Fieldbus, P-Bus, Foundation Fieldbus HSE, ControlNet, and others, most whose major market share is in the European Union), one would think that is all there would be to that. Unfortunately, as we have alluded to previously, these fieldbuses do not interoperate. Most are fundamentally different even to the Physical layers, as you may have ascertained from some of the networks described in this chapter.

Industrial Networks and Fieldbuses Summary

This has been a long chapter. It was almost summarized in the preceding paragraphs on Ethernet. Yet the point still needs making that these industrial networks attempt to meet the same goals. Some try to be overly broad, and some are dedicated only to process control. Today, it takes a network guy and a process guy just to determine the design of a new industrial network. Vendors will give you a turnkey package (the margins are much higher for them), but then you are locked into a proprietary solution that may work correctly for some or even all of the combined processes in a facility, however, you are at the mercy of the marketplace, vendors' whims, and changing requirements. If you go it alone and build a network out of components, it will allow you to control, but it will cost you in time and personnel. A standard industrial bus would certainly cut down on the time spent determining if everything is compatible and will work together.

Bibliography

Note that Internet links may change.

David R. Boggs, Jeffery C. Mogul, and Christopher A. Kent. *Measured Capacity of an Ethernet: Myths and Reality*. January 1995 ACM SIGCOMM Computer Communication Review, Volume 25 Issue 1.

Byrnes, Eric. *Instructors Notebook ISA Course FG21C*. Research Triangle Park: ISA, 2000.

—. *Instructors Notebook ISA Course FG30C*. Research Triangle Park: ISA, 2003.

Edison, John, and Wesley Cole. "Ethernet Rules Closed Loop System." *Intech 45*, no. 6 (June 1998): pgs 39-42

Fieldbus Foundation (organization website). http://www.fieldbus.org.

Fisher-Rosemount, Inc. *Fieldbus Technical Review*. Austin, TX: Fisher-Rosemount, 1998.

GGH Marketing Communications, Ltd. *The Industrial Ethernet Book*. http://ethernet.industrial-networking.com

OPC Foundation (organization website). http://www.opcfoundation.org.

Pinto, Jim. "The Great Fieldbus Debate—Is Over!"
http://www.jimpinto.com/writings/debate.html.

"Principles of P-Net." The P-Net Fieldbus. http://www.p-net.dk/booklet/bookpg04.html.

Rockwell Automation, Inc. Product Catalogs. http://www.ab.com/catalogs.

Smar International Corp. (corporate website). http://www.smar.com.

Telemechanique. Modicon PLCs. http://www.us.telemecanique.com/products/Automation/Programmable_Controllers/ index.html.

[1]Reviewer's comments on early DCS architectures.

7
Wide Area Networks

This chapter discusses communications over media and/or infrastructure that is not owned by the entity doing the communicating. That is, if you rent, lease, or otherwise do not own the media carrying your data it is part of the "wide area network" (WAN). This network may include the public switched network or leased portions of the network for private lines. Transmission over a wide area network is almost always performed in bit serial fashion (one bit after another).

Why should someone primarily interested in instrument loops care about wide area use or technologies? Because media characteristics haven't changed. Although the modems and digital lines described here may not be part of your immediate network, when a local area network wants to communicate with the outside world (and in the enterprise scheme of things chances are it will), it will do so by using one of the wide area technologies described in this chapter. Or, at a minimum, it will employ a method that can trace its parentage back to one of these technologies.

In discussing wide area networks, this chapter proceeds in an almost chronological fashion, beginning with wireline and wireline modems (including a brief explanation of modulation—and we do mean brief) and working up to the digital line offerings. A careful read will give you a good idea of where many of the serial standards (EIA 232 specifically) had their start and why.

New students of data communications may wonder why wireline modems were first used and why they were so slow (particularly if they remember their own experience with the Internet and 56 Kbps modems). First, we need to look back to the early 1960s. What network could then be found in almost all businesses? The public switched telephone network, of course. As you may notice as you read this chapter, a telephone line is not necessarily an ideal data path. Still, a telephone line existed in most businesses, was relatively inexpensive (compared to the then alternatives), and could be used for data transmission.

The author remembers quite well listening to a Bell Systems engineer in the mid-1960s who stated that data could not and would never be transmitted down a voice-grade line (your standard telephone line) any faster than 2,400 bits per second. As this is written, 56 Kbps (duplex) is a standard (V.92), and then there are the DSL lines (using that same copper pair) that exceed 1 Mbps. How was this increase accomplished? And what data transmission method should you choose now? Those questions are what this chapter is all about.

Keep in mind that control systems use LANs and are not normally run over a WAN. However, many supervisory control and data acquisition systems (SCADA) do use remote inputs; many are obtained over a WAN and even a wireline. And as the enterprise communications effort broadens, the need to interface to users who are geographically remote will become essential for any complete control system. One last thought before you start this chapter—many of the LAN developments used routinely in industrial data communications (such as TCP/IP) had their start in the wide area network, where limited bandwidth, noisy media, and the constant need for improved data speeds are the norm.

Wireline Transmission

This section describes WAN devices that fit into the first two layers of the ISO OSI model: Layer 1, the Physical layer, and (above it) Layer 2, the Data Link layer. These are sometimes referred to as "levels" in texts written before the creation of the OSI model. Layer 1, the Physical layer, provides the actual connection—the electrical and mechanical means to establish, maintain, and end physical connections between Data Link points. Layer 1 provides the functional and procedural means including "handshakes." Layer 2, the Data Link layer, provides functional means to establish, maintain, and disconnect data link connections between data terminals, modems, gateways, and the like. Please understand that there may be very little demarcation between Layer 1 and Layer 2, particularly if both functions are generated in a single piece of equipment, as may be the case with modems that are integrated into servers, and so on.

Most long-distance transmission of data is serial (in fact, almost all high-speed data is serial except for very local areas like a motherboard). This method makes possible two major types of links: switched and permanent. A switched network is much like the landline telephone system. The connections are originated at the beginning of a communication (off-hook and dial) and taken down at the end of communication (hang-up). A permanent connection is normally leased twenty-four hours by seven days a week. It utilizes the same media but doesn't go through the switching equipment.

Since the distance of the line introduces distortions and line losses to signals that have a large direct current (DC) component (known as baseband signals), an alternating current (AC) is used as a carrier for the data. This is accomplished through the process of modulation and demodulation, which provides a method for translating frequencies by taking the DC signaling rate and transforming it into a higher frequency range, thus eliminating the DC component.

Modulation is a term used to describe the "impression of intelligence on a carrier." In other words, modulation consists of modifying some sort of "carrier" (in this case, an electrical waveform) with some sort of information. This information is the "intelligence." Demodulation is the reverse, removing intelligence from the carrier. It is actually the more difficult of the two processes.

Many, many books have been written about both modulation and demodulation. The following information is therefore but a concise simplification of modulation and its effects on circuit behaviors. We will detail only the effects; the mathematical models are absent. Since almost all electrical communications of any great distance or high speed use one or more modulation techniques, grasping them is essential to understanding industrial applications in which data is normally transferred some distance. However, this discussion won't help you when selecting a communications device as that choice is made by the equipment designer, normally not the user.

Carrier Concepts

Why is modulation needed? Though there are many reasons, for our purposes here there's only one: to translate (move) a given signal's information to a different frequency. Why this is necessary will become evident in the discussion that follows.

Wireline Effects on a DC Signal

Normally, people are inclined to think of data communications through a wire as being instantaneous. This is because their observations are usually based on only short lengths of wire and are focused on such effects as turning on a car's headlights, which seem to come on "immediately." However, the transmission time from the moment the switch contacts are made until current is flowing through the lamp filaments is definable. This is particularly true of signals that have a fast rise (or fall) time such as data signals. Figure 7-1 illustrates the waveforms, both ideal and real, for a simple-series circuit that consists of a load (resistor), a source (battery), a switch, and a relatively long length of connecting wire. The switch's ideal output is an abrupt change (called a step voltage). After a long length of wire the end result is something less than abrupt.

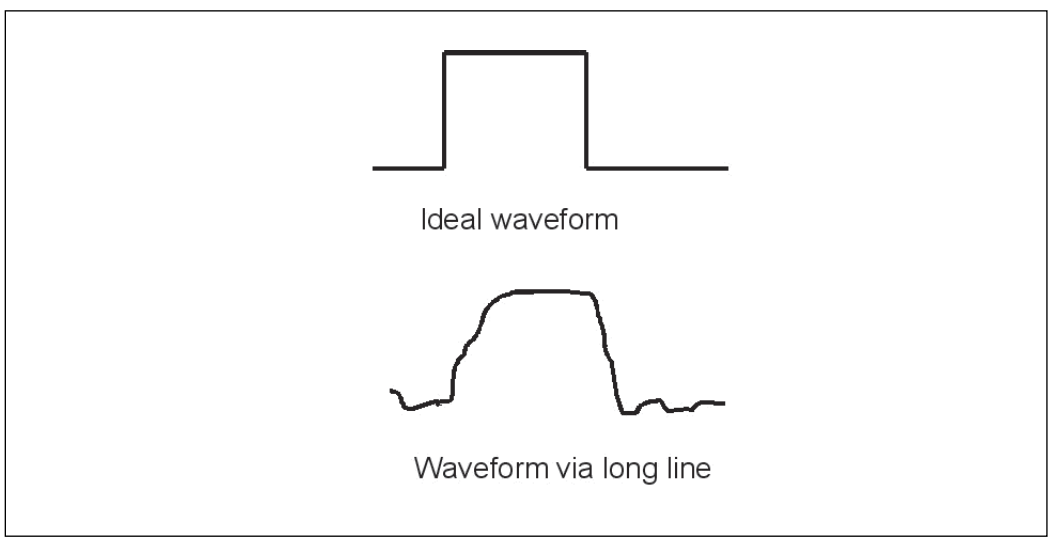


Figure 7-1. Wireline Waveform Changes

Note: The following discussion of wireline changes involves basic electrical knowledge. Skipping it will dilute your understanding of why modems are necessary but not what they do. So if you have a limited electrical background (or none at all) you may want to skip to the next section, "Sine Wave As a Carrier."

To see the reason for this output change, we must analyze the step voltage and understand that any conductor, such as a wire, has an inductance and a capacitance and that the longer the wire is, the greater the inductive and capacitive effects will be. Capacitance is parallel (shunt) between the conductors. Figure 7-2 illustrates the lumped-constant schematic. Lumped merely means that the inductance is distributed evenly throughout the conductor. The state of being lumped is represented by the schematic symbol for an inductor whose value represents the total inductance throughout the wire; the same is true for the capacitor, a single symbol representing the lumped capacitance of a segment of line.

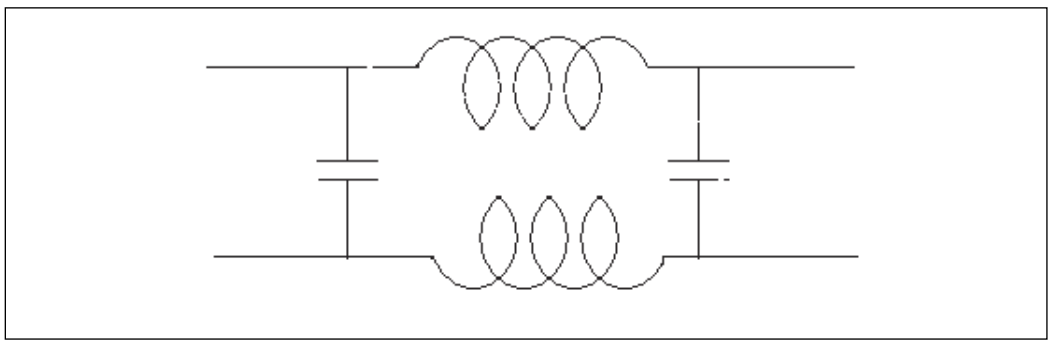


Figure 7-2. Schematic of Wireline

Observing the circuit that results from representing the lumped components shows it to be a PI (looks like the PI symbol π) filter with multiple sections. Arranged as shown in figure 7-2, it is a low-pass filter, meaning, of course, that high frequencies are attenuated, while the lower frequencies have less attenuation.

Analyzing the signal of a unit step waveform is complex and involves the use of higher mathematics. However, it should be easy to see that the unit step signal (the abrupt rise or fall) can represent any on-off signal that periodically changes state, such as a square wave. A square wave itself can be thought of as a sine wave whose period is the sum of one on and one off state or two element times. The square wave's leading and falling edges for the most part consist of the higher-frequency components. The level state is made up of the lower-frequency components—primarily the fundamental sine wave. Developing a square wave from the fundamental sine wave and its odd harmonics is performed by Fourier analysis techniques (quite beyond the scope of this text) all that is necessary to know is that the resultant square wave shape is formed by its low- and high-frequency components.

When a unit step voltage change or any other large swing in voltage levels occurs and a device attempts to transmit these changes down a long metallic wireline, the line's lumped capacitance and inductance and the line's copper losses (the DC resistance) combine to reduce the output waveform's amplitude and attenuate its higher-frequency components. The faster the rise time (or fall time) is, the greater the attenuation. Also, the metallic line acts as a delay line. That is, the reactive time constants are different for different frequency components of the square wave. This results in different parts of the waveform arriving at different times. The net result is a distorted waveform. The attenuation for low- or high-frequency components is called "amplitude distortion," and the different time constants result in an effect called "phase distortion."

The number of decisions that a media can support in one second is called the "line modulation rate" or "baud rate." You may determine a media's required baud rate by taking the smallest element or symbol time that you wish to transmit (the quickest decision time) and dividing this time into 1. A standard telephone wireline of 300 to 3,300 Hz is generally capable of 1,200 baud. That is twelve hundred decisions a second. For duplex operation this is six hundred decisions in both directions (adding up to twelve hundred). This is the maximum that the media can support.

Sine Wave As a Carrier

The sine wave is used as a carrier because a sine wave cannot be "integrated" or "differentiated" mathematically. Long wireline processes affect a square wave exactly as they affect a process controller. Integration (the averaging of change over time) causes a square wave to become a triangular waveform, while differentiation (the rate of change over time) on a square wave causes it to become a peaked waveform. These processes do not affect a sine waveform. Its amplitude can be reduced and the entire waveform shifted in phase, but the sine wave's shape cannot be changed through a linear or passive device. The fact that a sine wave cannot be integrated or differentiated by passive devices, that a square wave is made up of a fundamental sine wave and its odd harmonics, and that an electrical line acts as a low pass filter are actual physical phenomena, meaning, of course, that we may not necessarily know why, but we can use the rules. Because a sine wave will retain its physical waveform through linear processes it is used to "carry" digital information.

Modulation Process

Modulation is the process whereby a "carrier's" characteristic is modified to contain the information that is to be transmitted. The sine wave has three characteristics that may be modified:

(1) amplitude,
(2) phase,
(3) frequency.

There are many ways to impress intelligence on a waveform. Actually, even the lack of any change is "information." The first "modulation" schemes were merely turning the carrier "on" and "off." The on and off states are comparable to the dots and dashes of the Morse

code scheme. It is important that you not confuse "coding" and "modulation." Modulation is the altering of a carrier's characteristics, while coding is a scheme for the information itself. Over time, the on-off modulation of a carrier evolved into another form of modulation called "frequency shift keying," which is a form of amplitude modulation (not frequency modulation). In frequency shift keying, two tones are employed: one for a "1" and one for a "0." An example would be the HART protocol, which uses 1,200 Hz for a 1 and 2,200 Hz for a zero.

Amplitude modulation and frequency modulation, known as AM and FM, respectively, are familiar to most people through their radios and home stereo system receivers. AM is generally thought of as rather noisy and limited in frequency response but able to be transmitted over great distances. FM is considered quiet and capable of providing good frequency response, but unable to travel well, even from one city to another. These generalizations, while accurate, have less to do with the modulation than with the frequency range in which the two different modulations exist.

Probably less familiar is the nature of the TV signal. An American broadcast television signal has a 6 MHz baseband whose range covers 0 to 6 MHz. The sound is a subcarrier of 4.5 MHz, which is frequency modulated. The color information is quadrature modulated (independent sideband AM—ISBAM) with a 3.58 MHz reference carrier. This entire signal (sound plus color) is impressed upon the video carrier by vestigial sideband amplitude modulation (VSBAM). Today, even more features are put into the baseband signal, such as text and stereo sound. The modern TV (along with the VCR) is quite probably one of the most technically sophisticated devices in the average household, and we haven't even approached High Definition TV (HDTV) which is another layer of complexity. All of the broadcast modulation techniques just mentioned have been used to transmit data at one time or another.

Another reason for the use of modulation techniques involves the resistive losses in a long wireline. A DC voltage traveling through the line resistance will dissipate energy in the form of heat. Although the input voltage may be raised to compensate at the output, the higher voltage causes even greater energy loss. Alternating current can use transformers to match impedances and to select the correct voltage/amperage ratio for transmission. Amplifying an AC signal is easier than is a DC signal, so smaller amounts of transmitted power are required. Very seldom is a DC signal transmitted long distances over a wireline.

Before it's modulated or encoded a digital data signal has a maximum frequency (alternating ones and zeros) that is one half the bit-per-second (bps) rate and a minimum frequency that is zero when transmitting steady ones or zeros, which at 0 Hz is a direct current (DC). DC is not coupled across a transformer (no lines of flux being changed), and a voice-grade wireline has at least two transformers, one at each end. It should be obvious that we will have to change the DC component of our data signal if we want to transmit this signal at a fast data rate any distance at all (usually more than 100 meters) over a voice-grade wireline.

Amplitude Modulation

Amplitude modulation is the process whereby the amplitude of a carrier wave (usually a sine wave) is varied according to the information that is to be transmitted. Amplitude modulation involves more than simply turning the carrier on or off, although that in itself is a legitimate form of modulation. To better illustrate amplitude modulation, we will use the classic case: where the carrier is a sine wave at some frequency many times higher than the frequency of the modulating signal, which is also a sine wave. Figure 7-3 illustrates the frequency spectrum occupied as a result of modulation. The result may be graphically plotted, or a trigonometric identity may be used. In either case the outcome is the same. The output actually consists of four different frequencies after modulation: the carrier frequency, the modulating frequency, and two new frequencies, which are the result of modulation:

(1) the instantaneous sum of the carrier and the modulating frequency,
(2) the instantaneous difference of the carrier and the modulating frequency.

Figure 7-3 shows the results of a carrier at 1 MHz and a modulating frequency of 1 KHz. The sum and difference frequencies are then 1.001 MHz and 0.999 MHz and are called "side frequencies." It is these new carriers that actually contain the intelligence. The change in the carrier's amplitude at the modulating frequency rate caused the side frequencies.

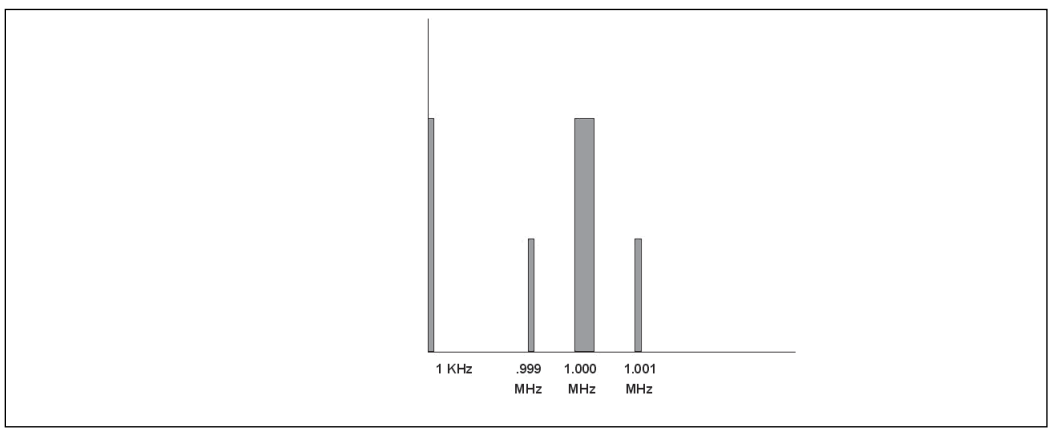


Figure 7-3. Carrier and Sidebands

It is important to note that total signal (carrier ±side frequencies) occupies 0.999 MHz to 1.001 MHz. If the receiving bandwidth is less than this, the information cannot be received (demodulated) correctly. Bandwidth is the span of frequency in Hertz; a signal occupies and is determined by the difference between the highest- and lowest-frequency components that are significant. In this case, subtract the lowest frequency component (0.999 MHz) from the highest frequency component (1.001 MHz) (determined the same as range is used in instrumentation) equals 0.002 MHz, which is 2 KHz (determined the same as span in instrumentation). If the modulating frequency is a band of frequencies, say, 20 Hz to 5 KHz, then the resulting required output bandwidth is 10 KHz, with each band occupying a 0.02-to-5 KHz range on each side of the carrier. This is known as "double sideband amplitude

modulation" (DSBAM). The 1.0 MHz carrier used as an example corresponds to the 1.0 or 100 position on the AM radio dial, and the sideband frequencies correspond (roughly) to those that the radio receives. In contemporary data transmission applications, the carrier frequency is not very high because the losses in a twisted-wire pair (as used in the bulk of installed telephone systems today) would not be economical. Wirelines (normal telephone lines) have a nominal frequency response of 0.3 to 3.3 KHz. If the signal had a frequency range of 20 Hz to 5 KHz while the wireline passes only 300 to 3,300 Hz, parts of the signal bandwidth will be lost due to attenuation of the frequencies not passed by the wireline.

When AM is used for data transmission over a wireline, the carrier frequency is a lot lower than our classic case. The carrier frequency is usually just above the signaling rate. Double sideband AM can be used for data transmission. However, DSBAM requires two cycles of bandwidth for every one cycle of the modulation frequency. This waste of bandwidth may be offset by the fact that it is far simpler to detect and demodulate DSBAM than in most other schemes.

Vestigial Sideband AM

The scheme most often used for AM data transmission is vestigial sideband amplitude modulation (VSBAM). This form is used because the bandwidth it requires is a little more than one half that required for a DSBAM signal. This is the reason why VSBAM is used in commercial broadcast television. The upper sideband is selected for television; however, either sideband could be used. In VSBAM the carrier is not fully suppressed, and a portion (a "vestige") of the upper sideband (for data transmission) is transmitted. It is important that some carrier frequency be received at detection so the signal will be demodulated correctly and in phase. (This is necessary because of filter characteristics; filters do not abruptly cut off at a certain frequency but have certain "roll-off" characteristics.)

Phase is not terribly important to analog voice users. However, for digital signals it is very important. Phase distortion is caused by the reactive components of the media and gives different parts of the signal different delays. Parts of the signal are not supposed to arrive later than other parts, even when all parts started at the same time. This distortion makes detection very difficult.

Single Sideband AM

Many people are familiar with single sideband AM (SSBAM) from its use in amateur and Citizens Band radios. A similar use in telecommunications is the independent sideband AM (ISBAM) signal system. The sidebands of a DSBAM signal are redundant (contain the same information). Therefore, if the carrier signal is totally removed (the intelligence is in the sidebands) and one sideband is eliminated along with the carrier, then all the transmit power can be employed in only one sideband. This process is known as "single sideband transmission." It does conserve bandwidth, but the receiver must be more complex. For demodulation to take place, the receiver must generate a signal of the same frequency as the carrier, which is used to demodulate the sideband. Though this modulation is quite

readily done today, the phase of the receiver-generated carrier is not always the same as the carrier that performed the modulation. This is what gives SSBAM that peculiar voice quality: the difficulty in correctly reproducing the carrier in the correct phase at the receiver. SSBAM is seldom used as the sole modulation method for digital signals. It is used in analog multiplexing schemes and in the radio transmission of data after the data has been converted from its "baseband" frequency by some other modulation method.

Independent Sideband AM

Independent sideband AM is a single sideband type of signal except that both the upper and lower sidebands are transmitted with the carrier suppressed. Each sideband carries independent (from the other sideband) information. In fact, the standard analog ISBAM transmission has four 3 KHz channels. This multi-channel scheme is not directly used in data transmission. An important characteristic of ISBAM is that when the upper sideband is used for one information channel and a lower sideband for another, they cannot demodulate each other. If they could it would result in distortion. You will see this technique used in Carrierless Amplitude and Phase modulation (CAP), which is used in digital subscriber line transport (DSL).

Frequency Shift Keying

One of the most popular and oldest methods of using modulation by digital data is the frequency shift keying (FSK) method. This method has two (or more) tones or carrier frequencies in the audio range: (1) a tone is keyed ON to represent a 1 state (A); the other tone (B) is OFF, and (2) the B tone is keyed ON to represent a zero state; the A tone is OFF. A rule-of-thumb requirement for this system is that the tones' frequencies should be separated by approximately the same number of Hertz as the signal's bps (bits per second) rate. As an example, if data is to be transmitted at 1,200 bps, then two tones, 1,200 and 2,400 Hz, would satisfy this rule. Actual practice allows less separation, but intersymbol noise will increase. The bandwidth of a typical FSK signal is illustrated in figure 7-4.

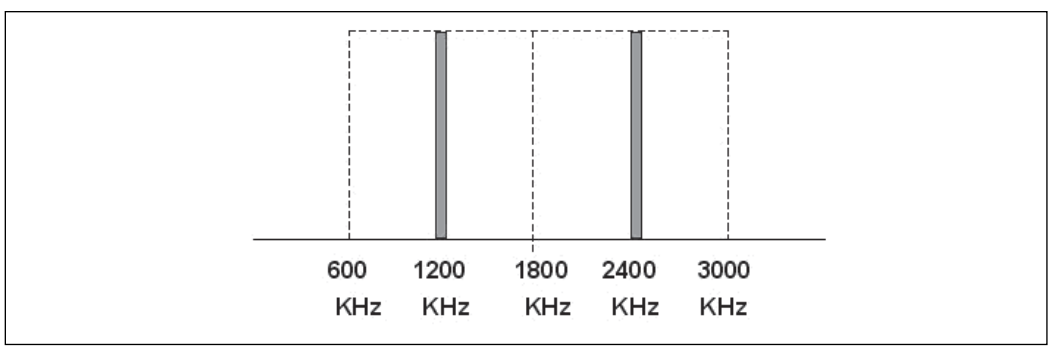


Figure 7-4. Bandwidth of a Typical FSK Channel

Why the rule of thumb? Why couldn't a system have two tones—for example, 200 Hz apart, keyed on or off at 1,200 bps? It could, but the signal could not be detected correctly. The data signal's maximum base frequency is 600 Hz. (The fastest rate of data change is

alternate 1s and zeros: if each bit occupies 1/1,200 of a second and a zero follows a 1 bit, then you have a cycle [two alternations] occurring in one six-hundredth of a second, or 600 Hz). A 1,200 Hz signal modulated at 600 Hz has a bandwidth of 600 Hz to 1,800 Hz. A 2,400 Hz signal modulated at 600 Hz has a bandwidth of 1,800 Hz to 3,000 Hz. If the sidebands overlap they will tend to demodulate each other, producing a signal that bears little resemblance to the original.

Frequency shift keying is employed because it is an easy process to use and fits in very easily with a binary digital transmission scheme. Even the frequency shift control is just a matter of a few logic gates. FSK, however, occupies two cycles of bandwidth for every bit per second. A 1,200 bps signal requires 2,400 Hz of bandwidth. Nonetheless, this is the method used in all low-speed modems (300 bps and less). It is also the method used for HART, in which a single 1,200 bps signal occupies 2,400 Hz of bandwidth.

The duobinary method is used to reduce the required bandwidth. It is an encoding method that does not allow a direct transition from one frequency to the other. It causes the output to transmit a full cycle of the upper FSK tone and a half cycle of the lower FSK whenever a data transition occurs. To the line this appears to be a 600 Hz signal that uses tones at 1,200 Hz and 2,400 Hz. Using duobinary causes three tones to be output: 1,200 Hz, 2,400 Hz, and, if an alternating digital signal is used (fastest rate of change for a binary signal), 1,800 Hz. This last output, 1,800 Hz, is the average frequency output because duobinary does not allow a direct transition from one frequency to the other. This technique gives one cycle of required bandwidth for one bit time.

Another method, known as biphase, but more popularly as Manchester encoding, places the state of the binary signal into a transition. If the signal transitions from a zero to a 1 it represents a zero state. Transitioning from a 1 to a zero represents a 1 state; a return to the opposite state will be at clock (twice the data rate) and is ignored. The net effect for a data signal that is alternating 1s and zeros is the same output as duobinary, that is, three frequency components. Manchester encoding is used quite often at speeds of up to 1000 Mbps.

Frequency Modulation

In frequency modulation, the modulating signal varies the carrier's frequency change from rest according to the modulating signal's amplitude. Moreover, the carrier's frequency changes vary at the modulating frequency. An important fact to remember about FM is that the output carrier's amplitude remains constant; only the frequency changes. Most noise involves amplitude; that is, it tends to ride on signals above and below that of the average carrier. By slicing off the top and bottom of the received carrier (a process known as limiting), most of the amplitude-type noise can be removed from the FM signal. All this is not without cost, however. Bandwidth is needed to properly demodulate an FM signal. There are two kinds of FM: wideband and narrowband. Most of FM's appealing characteristics are inherent only in the wideband type. An important fact to remember about FM is that the

"deviation"—the amount by which the carrier frequency is varied—depends only on the modulating signal's amplitude. The frequency of the modulating signal determines at what rate the deviation takes effect, but not the amount of deviation.

An FM signal has an infinite number of sidebands. They are spread out over the frequency spectrum, above and below the carrier frequency at differing amplitudes. Their relationship to the carrier is such that each sideband is an integral (whole number) multiple of the modulating frequency away from the carrier. That is, if the modulating frequency is represented by "(fm)," then the sidebands are separated from the carrier frequency by: ±fm, ±2fm ... The amount of energy in each sideband decreases as its distance from the carrier increases, eventually becoming insignificant. The decrease is not characterized by a simple linear relationship, but rather is based on a mathematical treatment using Bissell functions. This treatment is beyond the scope of this discussion, but may be found in any standard text on frequency modulation. What is an FM signal's bandwidth requirement? Typically, without encoding, three cycles of bandwidth are required for each bit per second of data.

We stated earlier that there were two kinds of FM, wideband and narrowband. The dividing point between them is their deviation ratio or the ratio of the maximum carrier deviation divided by the maximum modulating frequency. As an example, if the maximum deviation is 75 KHz and the maximum modulating frequency is 15 KHz, then (75 ÷ 15 = 5) the deviation ratio is 5. Signals that have a deviation ratio greater than three are considered wideband; if less than three, they are considered narrowband. Because the FM signal requires a large bandwidth (in uncoded form), it is seldom used in digital transmission as the sole means of modulation.

Phase Modulation

Essentially, phase modulation is just like frequency modulation, except that the amplitude of the modulating signal causes a shift in the reference (or center) carrier phase. The difference between FM and phase modulation (PM) is this: for a given modulation signal amplitude, phase varies only with the modulating signal's amplitude and not the modulating signal's frequency. FM, on the other hand, varies directly with the modulating signal's amplitude but inversely with the modulating signal's frequency. In other words, using FM, the lower the frequency is, the greater will be the deviation for a given amplitude. That is all (from a practical aspect) there is in the way of theoretical difference. It is possible to obtain FM from a PM generation or PM from an FM generation; it is all in how the modulating signal is presented. Please note that a continuing change in phase is a change in frequency.

Encoding Data

To increase the data rate for a particular modulation scheme without increasing the line's baud rate requirements (baud is the line modulation rate), the digital data may be encoded. One of the earliest examples of encoding is the process called di-bit encoding. In it, a buffer holds the input data to be transmitted, and decisions are made on every two bits. As an

example, use 2,400 bps. Logic is constructed so a decision is made every 1/1,200th of a second rather than at the bit rate of 1/2,400th a second. Examining the bits in pairs is the reason for this decrease in decision-making frequency. Four possible combinations will be possible from the two binary digits, but a decision about which one of the four combinations is needed only has to be made 1,200 times a second rather than 2,400 times a second. This is now a four-state signal. Figure 7-5 illustrates the phase modulation technique that is used to transmit the four states.

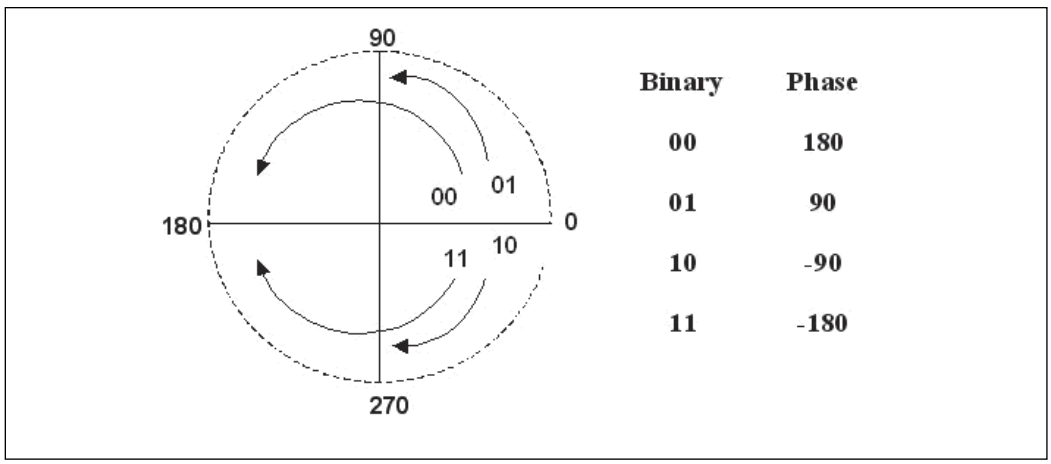


Figure 7-5. Quaternary Phase Shift

This is type of phase modulation is called continuous phase modulation or continuous phase shift keying (PSK). The last transmitted phase becomes the reference for the next transmitted phase. Obviously, a long string of 1s or zeros could cause a continuous change in phase, and the transmit frequency would change. If the change is significant enough the receiver will lose synchronization. This can be remedied by inserting circuitry and logic that causes predictable state changes at the modulator and removing the changes upon demodulation; this circuitry is known as a scrambler.

The resulting four state modulation is referred to as "quaternary phase shift keying" (QPSK). It reduces the line bandwidth required for a given data rate signal. Of course, the laws of physics won't give something for nothing, and in this case the something is signal-to-noise ratio. The QPSK method requires a signal-to-noise ratio that is higher than that of the uncoded signal. At the demodulator, the last received phase becomes the new zero reference phase. This phenomenon is called "differentially coherent detection." Multiple-bit systems form the basis for modern data transmission technology.

Tri-bit systems have been used that have eight different phase positions. Newer techniques allow five or six bits to be encoded. A selection is then made from thirty-two (five-bit) or sixty-four (six-bit) data states and choosing the phase change that makes the largest change of phase for the bit combination being encoded. This technique is called "trellis-coded"

modulation; so called because when the possible states are graphically located the resulting image is that of a trellis. Trellis-coded modulation requires the use of a selection algorithm, which is usually executed by a microprocessor. In most "standard" modems, not only are phase shifts selected; the amplitude modulation of the particular phase is also selected (typically one half or full power). This results in a particularly large trellis. Trellis-coded modulation is how 33.6 Kbps duplex (a total of 67.2 Kbps for both directions) makes it down a 1,200-baud wireline. The number of bits encoded fools the line into thinking only six hundred decisions a second were made in each direction. Figure 7-6 is a representation of trellis coding, showing the first 90° of phase difference. Twelve decisions are possible: six for each 15° change in phase shifts and another six regarding whether they are at half or full power. For four quadrants this means forty-eight possible states. Typically, five bits are used; this would give thirty-two actual states required, and thirty-two times six hundred (the allowable decisions made in one direction) would give a data rate of 19,200 bps. There are more combinations than decisions because the algorithm always chooses the maximum deviation for that combination. The receive demodulator must have the same algorithm.

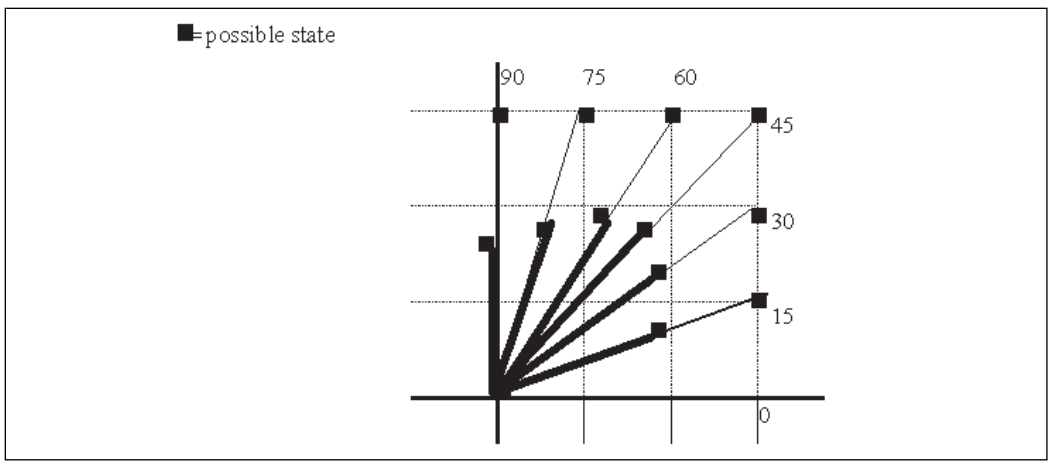


Figure 7-6. Trellis Coding

The point of this discussion should be to emphasize the importance of having identical selection algorithms at each end; without them, detection will be impossible.

Demodulation is the reverse of modulation. Once the phase combination is recovered then that particular combination of bits is placed in the output register and transmitted at the bit rate. Note that you may modulate information as fast as you want. If it cannot be demodulated then you cannot use it; it is just that simple.

Summary: Modulation

Only three components of a sine wave may be modulated: the amplitude, frequency, and phase. In fact, frequency modulation and phase modulation are usually lumped together as "angle" modulation. Amplitude modulation changes the carrier's amplitude according to

the modulating signal's amplitude, and the changes in the carrier are at the modulating signal's frequency. Frequency and phase modulation change the carrier's frequency (phase), based on the modulating frequency's amplitude. These carrier changes take place at the modulating signal's frequency. Though the basic modulations have certain line requirements, encoding the data effectively makes it possible for transmission lines to have higher data rates.

Wireline Modems

Serial modems originated in the 1960s with point-to-point serial data transmission, which, at the time, was the most economical way to transmit data over long distances. In the terms of network architecture, a typical 1960s' computer network block diagram was a "star" type—the same architecture which is used with direct digital control (DDC) in instrumentation systems. In the star architecture the main computer is the hub. All things are either transmitted to or from the hub; no lateral transmissions are permitted. Most loops are local, though several could operate through modems. Note that if you replace the word *hub or main computer* with *Internet* you will have the world as envisioned by those who are net-centric.

Over the years, some things in the modem world became more or less standardized. The modem's parameters were expressed in terms of the (predivestiture) AT&T (Bell System) specifications. These specifications are both historical and contemporary. Long-distance data communication, although more rapid than two decades ago, still uses some of the same specifications as well as all the old technical jargon.

Definitions

Let's start our discussion of modems by defining key terms and the technical rationale for them.

Asynchronous

Generally, *asynchronous* means that it may occur at any time and is not tied to a clock. The old "start-stop" teletypewriter signal with its one start bit and one (1.45 or 2) stop bit(s) is a good example. This signal started out using motor speed as the main synchronizing element, and the start-stop bits synchronized each character. In today's vernacular, any start-stop signal is assumed to be asynchronous.

Synchronous

Synchronous generally means tied to a common clock, the clock signal being transmitted along with the data. Originally meaning a signal that had no need for start-stop synchronization, *synchronous* used bit timing so each *bit of data was accounted for*.

Baud Rate and Bits per Second

Baud rate and bits per second are often used interchangeably, but this usage is incorrect. A baud is a line modulation rate, that is, what the transmission media needs in order to pass

the data. Line rate in bits per second is the transmission speed of the device that is transmitting or that the device is capable of receiving. The bit rate (throughput) of a modem depends on its line rate, the condition of the line, the packetizing method used, and the amount of data compression. *Baud per second is* a term for describing a change in transmission requirement, not a line speed. An ad for a V.90 56 Kbps modem (the line data rate) is for a modem that requires a 1200-baud line, that is, a line that has a bandwidth great enough for 0.0008333-second rectangular pulses.

Preamble

A preamble is a physical layer signal that is used to synchronize connected devices. In many systems, a burst of carrier (or clock for baseband) is sent to synchronize receivers. This is followed by bit patterns to synchronize the bit timing, then control patterns (characters or a unique bit arrangement) to synchronize messages.

Synchrony

Synchrony in the context of modems relates to the way two modems will get their bit-rate clocks in phase—or how they will know when they are in time with each other. Many modems will supply a "receive" or "recovered" clock. This may be used to synchronize (or even drive) the data terminal equipment (DTE). In most modern communications devices, the transmit signal's transitions will be performed at the transmitter's clock time so the transmitter's clock timing is inherent in the data signal's transitions. These may be recovered on the receive end and the receive DTE's clock adjusted until it is in phase with the transmitter's clock.

As one works through the various modems and networks available, a pattern emerges: first, frequency synchrony is achieved, then bit synchrony. Though methods differ, this will be the pattern in all synchronous systems. Generally, an asynchronous modem will not supply a recovered clock, and a synchronous must.

Legacy Modems

Wireline modems were for use over dial-up or leased telephone-type lines and are divided into two classes of specifications: Bell (the name Bell is generally no longer used, just the specification number to describe a modem) and the International Telegraph and Telephone Consultative Committee (CCITT), now ITU-ITS, which issues the "V." and "X." standards. Table 7-1 lists various standard modems with their speed and modulation types. Most modern modems are synchronous between the modems, using LAP-M packets, and may have a synchronous or asynchronous interface. On PC modems the interface will be asynchronous as the PC bus for I/O is asynchronous. Depending on price and application, stand-alone modems may be configurable for interface through either programming or the use of a dip switch.

Asynchronous Modems			
Speed	**Bell**	**CCITT**	**Modulation method**
300	103/113	V.21 (similar)	Frequency Shift Key
1200 (half duplex)	202	V.23 (similar)	Frequency Shift Key
1200 (duplex)	212A	V.22 (similar)	QPSK (di-bit)
Synchronous Modems (PC internal types are asynchronous input, but synchronous between modems)			
Speed	**Bell**	**CCITT**	**Modulation method**
1200	212A	V.22 (similar)	QPSK (di-bit)
2400	201	V.26 (similar)	QPSK (di-bit)
4800	208A	V.27 (similar)	OPSK (tri-bit)
9600	209A	V.29 (similar)	AM/PSK
9600		V.32	Trellis coded
14400		V.32bis	Trellis coded
19200		V.32terbo	Trellis coded
28800		V.34	Trellis coded
33600		V.34	Trellis coded
56K (download) 33.6 (upload)		V.90	Broadband on download, trellis coded on upload
56K (download) 48K (upload)		V.92	Broadband on download, trellis coded on upload

Table 7-1. Modem Characteristics

Typical of many of the low-speed legacy wireline modems is the Bell 212 type, which contains a Bell 103 set of carrier frequencies. A plug-in card for many PCs, this once was the most widely used modem on dial-up lines. The Bell 212 is capable of identifying what type of modem it is talking to and adjusting itself to become that type. Considering the great differences in modulation schemes, and so on, this is no small feat.

Faster Modems

The Bell 212 modem had to use data encoding to overcome the bandwidth restrictions of a typical dial-up wireline. An older specification, Bell 202, operated at 1,200 bps using FSK, with one carrier at 1,200 Hz and the other at 2,200 Hz. This scheme used almost all of the bandwidth available to the typical two-wire wireline and had to operate half duplex. This caused throughput problems because the predominant method of error recovery is ARQ (Automatic Retransmission Query), and after a number of transmitted blocks an acknowledgment must be returned to the transmitter. To avoid having to perform handshaking every time the line was turned around (in order to send an acknowledgment of message receipt), the 202 used a secondary channel. A small portion of the spectrum near the 300

Hz lower limit was used for a 5 bps channel. Though this could be called duplex, it is only technically so. The problem was the use of FSK as the scheme of modulation and the limited wireline bandwidth. The Bell 202 could actually be called the originator of the asymmetrical data line, in which data in one direction proceeded at a much higher line rate than data in the other direction.

Enter the 212 modem. It used the di-bit-encoded, quaternary phase shift keying with coherent detection. It is a synchronous modulation/demodulation scheme used because it has noise advantages over a four-level AM signal. The 212 has a unique answer tone that allows for the convenient identification of modem speed when the called end answers. This modem could be used in asynchronous or synchronous systems, offering the choice of recovered receive clock (or not). From the 212, modem types data speed progresses until you reach the V.90 modem. V.92 is not considered a legacy modem yet.

We have not mentioned error-detection and -correction schemes. Operating above 2,400 bps over a wireline presents opportunities for many more errors than at the slower speeds. Modem manufacturers therefore offer different data compression as well as error-detection and -correction schemes. Unless one is operating with identical schemes used in both modems, the error-detection and -correction process will generate errors. Two sets of standardized schemes are in general use. One scheme is the V.42 international standard, which in the original and bis (second modification) allows for data compression at two to one (original) and four to one (bis). The others would be the Microcom Networking Protocols (MNP), which come in various flavors, the first five having been released to the public domain. MNP-1 and MNP-2 are generally concerned with the packetizing of data, but MNP-3 through -5 get serious about a whole range of parameters, including losing the start and stop bits (as does V.42) for a 20 percent gain in throughput and packetizing the data. V.42 employs the LAP-M frame and uses code lists to represent strings of characters, replacing the strings with the shorter code lists. MNP-6 through -10 are still in the proprietary domain, meaning that you have to have modems at both ends that have the appropriate level of software.

How does a modem know what to do? Part of the modern modem technique is something called "training" or "negotiation." In this process, the two modems negotiate which speeds, compression, and error-detection schemes they will use.

Modems are used for frequency translation, and wireline types are only a small segment of the devices called modems. Others are used in local area networks (LANs) and cable television (CATV, which stands for Community Antenna Television, where cable has its roots). Still others are found in satellite and ground communications, just to name a few. Depending on the medium, modems use varying techniques to transmit information. There is a continual push to increase transmission speeds. Especially in the wireline arena, the speeds keep increasing. The highest-speed "standard" modems for wireline at the time this is

written are the V.90 and V.92 at 56 Kbps. However, because of the need for increased throughput over the wireline medium exists, efforts are being made to go past these speeds when economically possible. The V.92 standard improves slightly on the V.90 specifications by adding a method for disconnecting the modem long enough to let you know that someone is trying to call you without losing the connection, a feature referred to as "Internet call waiting." Also, the maximum upload speed has been increased from 33.6K to 48K.

Higher-speed wireline modems must use a variety of techniques to coax high speeds from a 3 KHz wireline. One of the techniques used prior to the V.32 modem series (and still used on hard-to-transmit cases) is multi-tone. In multi-tone, the data stream is divided into eight (or more) parallel streams. Each stream is di-bit encoded, after which each performs phase shift modulation on its own carrier (one of the eight or more tones separated by about twice the tone bandwidth in Hertz). All the tones are used as a simultaneous baseband to a vestigial sideband AM modulator. At the receive end, the reverse process takes place until the signal is reassembled. This scheme is the basis for DMT (discrete multi-tone), which is used in DSL lines.

Most modern high-speed modems now incorporate a "fallback" system in which, if the error-detection rate rises above a preset level, the speed falls back to a lesser data rate. Some drop in integrals of standard data rates, some by smaller increments. As the data error rate improves, the speed is adjusted back up to the desired rate. None of these newer modems would have been possible without the large advances in microcircuitry. What would have been literally tons of equipment is now routinely packaged in one integrated circuit (IC). It's interesting to note that high-speed modems typically start up at a low speed (1,200 bps). If transmission is not possible at 1,200, there is no use in trying higher speeds. If transmission is possible, they then negotiate the speed, data compression, and error-detection scheme the modem at each end will use, arriving at the highest common denominator of operation. This procedure is required because of all the different types and speeds of modems and the various error-detection and compression schemes available.

Summary: Modems

In this chapter we have discussed modems, from low speed to high speed, that operate over the typical telephone wireline. Data speeds increased from the high speed of the early 1970s (2,400 bps) to a magnitude or greater than that in just three and a half decades. Even more significant is the fact that for one of the first auto-dialing, auto-answer modems you would have paid more than $500 for a data rate of 300 bps in 1984, and in 2007 you could purchase a 56 Kbps modem with bells and whistles for around $20. And that is without considering data compression, which can up the throughput to a continuous average of near 2.8 to 1, nearly 250 times the performance for one tenth the costs, and these modems are more reliable. The explosion in the use of Internet graphics and the need to transmit objects rather than just character-based text have fueled the insatiable appetite for more bytes per buck and wringing the highest possible data rate out of the wireline.

Can speeds greater than today's wireline modems be increased? Remembering the engineer who once told the author, "No more than 2400 bps down a wireline…." the author refuses to speculate. I do know that for those who need more bandwidth now, there are present-day solutions but not necessarily at one-tenth the cost of earlier modems. These solutions are outlined in the next section on digital line offerings.

WAN Digital Lines

A wide area network (WAN) is commonly understood to be one that serves geographically separated areas. One of the best examples is the public telephone network. There are also wide area data networks. IEEE 802.6 standardizes Metropolitan Area Networks, which, in turn, can be connected together as WANs. Intercity links are usually 1.544 Mbps, the T1 carrier data rate, which has become the standard telco channel. Subscriber rates are available as fractional rates such as 56 Kbps, 64 Kbps, and so on.

Telephone Lines As Media

There are two problems with the bandwidth of the analog telephone line even at its best: (1) frequency selective amplitude distortion and (2) frequency selective delay (phase distortion). Since the ear is most vulnerable to amplitude distortion, most phone lines are corrected for the ear. However, because the ear is not very sensitive to delay distortion, many of the techniques used to reduce amplitude distortion increase delay distortion. For digital transmission, these conditions are corrected as much as possible through extra features called "conditioning," which may be focused in the line, in the modem, or possibly in both.

Higher-speed modems also have automatic conditioning that is based on pilot tones or bit error rate (BER). Rather than adding conditioning to analog lines, a subscriber can purchase or lease digital lines. For years, telephone operating companies have been selling digital lines to customers by leasing a line, a data service unit (DSU), and channel service unit (CSU). The DSU may be customer owned, but the rest of the transmission is the phone company's worry.

The source of data-ready lines is typically fractional T1 lines (explained later in this section). Remember, digital data will not transmit on analog telephone lines because it has a DC component, which is caused by successive ones or zeros. Most analog telephone lines have numerous transformers in the different terminations and couplings, so a frequency translator, such as a modem, must be used to put digital data onto these lines and recover it at the distant end. Optionally, the transformers (or more correctly the line-impedance matching devices) may be bypassed and the copper accessed directly. The typical modern voice-grade line by itself is capable (through differing techniques) of transmitting digital data at speeds above 56 Kbps for limited distances. Digital lines will support baseband digital. However, there should be some provision, either in the hardware or in the DSU, for eliminating long strings of one state or the other so that enough transitions take place to prevent loss of synchronization.

Direct Distance Dial (DDD)

The direct distance dial (DDD) system is used for calls outside the so-called local access and transport area (LATA). The ten-digit LATA number (in North America) is composed of a three-digit area-code prefix to the normal number, which is a three-digit central office and four-digit station number. Calls on LATA can be routed in many ways: by radio link, satellite, wireline, microwave, or fiber optics. All of these are digital trunks. In North America, only the subscriber line, the so-called last mile, is analog; all other transmissions are digital. So the PC is digital to the modem, converted to analog to go over the subscriber line to the central office (CO), and converted to digital for long lines. At the destination CO it is converted back to analog, where it is sent to the destination and converted back to digital. What a lot of wasted conversions.

Not all lines in the world are digital, however. Some long lines (outside of North America) are still analog and connect to trunk lines where the circuit is converted from a two-wire (one pair) to a four-wire circuit—one pair for transmit and one pair for receive. Why? Simple! Some lines need amplifiers to make up for loss. These amplifiers have only one direction, from the source to the destination, since each pair of a four wire line is unidirectional. A little study will show that if the hybrids (two- to four-wire conversion devices) or the digital terminations are not exactly balanced, then some of one line's energy will slip over into the other line's amp. If that happens at the other end also, then an oscillator is created that produces, at the very least, an echo that is very distracting to the person trying to talk or listen. To deal with this problem, an echo suppressor is placed in both lines. Actually, this is an electronic switch that allows one amplifier to be on at a time. Neither party can talk simultaneously and communicate. Normally, one person talks, and when he or she provides an opening, the other person talks (unless, of course, you are arguing with your significant other). The echo suppressor listens, and, when there is no energy in one line, it will turn the amplifier off. It takes only 100 milliseconds of silence (0.1 second) for the echo suppressor to switch one party off and the other talking party on. This action is not noticed in the normal course of human speech. Both *digital* and *analog* lines can have echo suppressors.

With a duplex modem, there is a problem: some lines have echo suppressors; others do not. Communication is not possible in both directions, as duplex operation requires, on lines that have echo suppressors because one of the amplifiers will be off. This is taken care of in the modem handshake. The answer tone will last in excess of 400 milliseconds. When the echo suppressor hears the answer tone for a sufficient length of time, it turns off the echo-suppressor feature until the next 100 milliseconds of silence occurs. Since the modem handshake, under normal circumstances, leads to tones in both directions until transmission is complete, the echo suppressors are effectively out of the circuit. If the line is dropped, even for only little more than 100 milliseconds, connection must be reestablished.

Packet Switching

Packet switching is the process of breaking up a digital information stream into packets. Each packet will have its destination (and source) address included in it, perhaps with control and data as well. Obviously, these items must be arranged in a specific format. CCITT (now ITU-ITS, International Telecommunications Union—International Telecommunications Standards) Recommendation X.25 concerns a packet-switching scheme to be used over the public network. It has specific controls and is generated in a Packet Assembly/Disassembly Device (PAD). X.25 uses a statistical multiplex technique that adds packet headers to the user's data. The PAD breaks a message up into small packets, each addressed to the destination, and inserts them into the transmission channel when it is available. Each step is connection oriented in that a Logical Channel Number (LCN) (recyclable) is assigned when the packet is transmitted from the originating station to the network cloud. See figure 7-6 for the X.25 packet switching layout. At the receive end of the cloud, the packets are then transmitted to the receiver, with another LCN assigned to them.

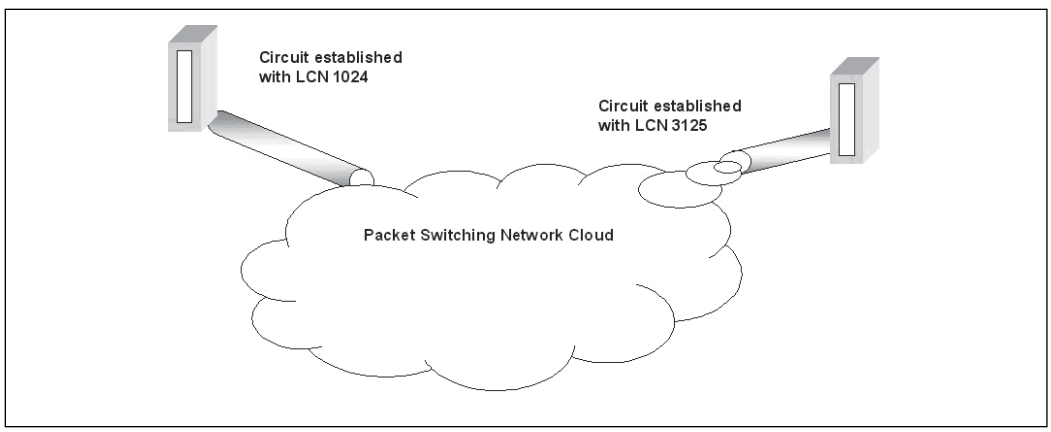


Figure 7-6. X.25 Packet Switching Layout

The receive end then reassembles the message from the packets. Various data speeds can be used; however, 56 Kbps is the highest, and others are a submultiple of 64 Kbps. Typically, only virtual circuit service is used, and that is a connection-oriented service.

To review, in connection-oriented transmissions, a destination is set up, and the packets are routed to that source in the same sequence in which they were generated, even if it means the packets must be stored at one point while they wait for an available path. Connectionless service allows the packets to be routed by the available path, and they may arrive at the destination out of sequence. It is the responsibility of the receiving device to reassemble the packets in the correct order. (This could be tough on real-time voice or television.)

X.25 uses an HDLC (point-to-point) frame in which a protocol data unit (PDU) is enclosed. Packet switching is performed on the public switched network, although it could doubtless

be used on private networks. X.25 is supposedly available everywhere (in theory) as a data network. And since these data packets can exist with a digital voice scheme, they become the basis for an integrated network of voice and data. Though the public switched network and packet switching itself are viable, proven technologies, they are not presently configured to meet the needs of industrial data communications in that they are not inherently deterministic. A small amount of time researching X.25 with an Internet search engine will reveal many good texts relating to X.25 and packet switching in general, *and the reader is referred to them for further investigation*.

Integrated Services Digital Network (ISDN)

ISDN service hoped to offer both voice and data, and perhaps video, all integrated into one service and generally distributed throughout the local area as long as the on–premises equipment was not more than 13,000 feet from the central office. The twisted wire that now carries telephone service was a limiting factor. Later, as fiber-optic loops fell in cost and became more readily available, a host of services were offered via the loop from your local telephone company. These could include all those now found on cable TV; those services now available to a computer user through a modem, plus video text and video telephony; and services just now coming into widespread use, like true video on demand or your TV connection to the Internet through a high-data-speed channel provided by either your local cable company or telephone operating company.

Though the technology to implement ISDN has been here (it was designed over twenty-five years ago), the implementation has lagged. Because they have a monopoly over telecommunications and can dictate standards, many foreign governments have already established standards for ISDN. Some of these standards clash with the technology and rationale of the competitive systems found in the United States. One might assume (at one's own risk) that the political compromises necessary to realize every international standard could be achieved. But they were not, and other, newer systems have displaced ISDN as the system for connecting all the businesses and residences in the United States. The objective of ISDN was point-to-point digital connectivity. For this to be accomplished, it was conceived using the sometimes analog public telephone network. One of the key features of ISDN is out-of-band signaling. This means that the control signals are in a different media than the messages.

Originally, there were two different ways to connect to an ISDN network: basic and primary. In the basic configuration, three channels are distributed on the wireline. These are 2 B (for Bearer) channels operating at 64 Kbps each, and 1 D (for Data) channel operating at 16 Kbps. This is called the 2B+D arrangement. The original intention was that the B channels could carry either digitized voice or data on either channel and that the D channel would provide control signals and other low-speed signaling requirements. You can tell at what technical generation ISDN was conceived by the fact that it took 64 Kbps for subscriber-quality voice. In modern digital telephone systems, 32 Kbps will give you toll-quality voice, and 16 Kbps will give you subscriber-quality voice.

Message traffic on the D channel uses Link Access Protocol-D (LAP-D), which is essentially HDLC. The difference between the two lies in the address fields, where LAP-D uses a two-octet address—one to identify its network, and the other the end point. By using the D channel to control the switching of the B channels, a clear channel is established in which bit patterns on the data (B) channel do not affect transmission. At the time of ISDN's conception 64 Kbps was chosen because that was the data rate needed to support voice digitization. That rate is quite high for voice given current technology, and in many cases is too low for some of the digital data services. It would be hard to use an ISDN line as a LAN port without having much reduced data rates (at least with the IEEE 802 types). Additionally, some of the electronic switches will not support an ISDN channel at 64 Kbps. Instead they require transmission at 56 Kbps because they extract the 16 Kbps signal by using 8 Kbps from each of the 64 Kbps channels. This results in two 56 Kbps channels and a 16 Kbps data channel that is in band for signal control. ISDN has achieved a degree of popularity, not as an integrated service medium, but for relatively fast Internet access.

In primary access, many subscribers use common input trunks to a facility. These trunks are multiplexed together into twenty-three B channels and one D channel and are able to operate across one T1 carrier line. An application of ISDN is the private branch exchange (PBX) facility, which is fed by the T1 trunk and distributed out to the subscribers.

Would ISDN be of interest to industrial users? It could provide a gateway between a LAN and the telephone WAN. However, on most LANs the data-signaling rates might cause a severe bottleneck in data transfer. Some ISDN terminal adapter vendors make a modem to utilize the entire bandwidth (128 Kbps, but you will only run at 115 Kbps). The author's experience with ISDN is that it is not quite ready for prime time unless you live in a major metropolitan area and all you want is fast access to the Internet. First, configuration is not simple. Second, many operating systems do not have drivers for an ISDN terminal adapter through the serial port. Worse, if you wish to connect via ISDN and are a user in a less-traveled location, more than four miles from the central office, you may experience more than just a few problems attempting your original installation. Due to the sharply rising demand, equipment may not be in the central office for a while. Your installation costs (depending upon the state and the tariffs imposed) vary widely. You are still subject to recurring line charges. Lastly, remember, another ISDN facility must lie at the other end of your transmission in order to make good use of the ISDN features. At present, these are not plentiful. One of the good ideas that came out of the original ISDN is frame relay, which we discuss next.

Frame Relay

Frame relay is the connectionless version of X.25, and instead of using LCN it uses a Data Link Connection Identifier (DLCI). The carrier company (or companies) provides Layer 1 and 2 services for your data. In other words, you have a frame relay connection into which you pump data and extract the same. Frame relay's speeds start out at (typically) 64 Kbps and can go as high as you have the pocket money to afford. Though frame relay in theory

offers switched circuits (like X.25), currently most connections are not switched but are rather permanent virtual circuits. These connections are point to point as far as the user is concerned, but the actual routing and number of nodes varies and is unknown to the user. The customer is given a Committed Information Rate (CIR), for which he or she pays a tariff. Normally, the user chooses the CIR at the average data rate of his or her traffic. As long as traffic is below the CIR, packets will not be discarded. As the data rate goes above the CIR, a best effort will be made to deliver the packets. The user's Layer 4 functions will discover the discarded packets. At some predetermined rate above the CIR all of the user's packets are discarded if there is congestion on the network. Discarded packets are the frame relay network's way of saying it is becoming congested. Frame relay may very well be used to connect corporate LANs, and the present-day cost per single channel is low enough for industrial usage, particularly if the frame relay DDS is used. Frame relay DDS provides 56 Kbps with no CIR, using the Digital Dataphone-type tariffs (actually fractional T1 now), and it provides this service typically at the monthly cost of a business telephone line.

T1 Carrier

T1 carrier in (North America) is a twenty-four-channel time division multiplexed channel. It multiplexes the outputs of twenty-four DS0 channels onto one 1.544 Mbps line, generally a coaxial cable but increasingly fiber optics. In the past, one had to buy all twenty-four channels (or 1.544 Mbps), and this was expensive. Due to the falling price of multiplex equipment and increased competition, fractional data rates can be bought from 64 Kbps up. Sixty-four kilobits per second is the single channel data rate (DS0) of the twenty-four channels that make up T1. Actually, in most cases the customer only gets 56 Kbps since the provider will steal one bit from each DS0 for control. A T1 channel is also known as a DS1 channel. In the Extended Super Frame method of signaling, there are twenty-four DS0 channels (with 193 total bits: 192 for data and 1 for framing). Since the original voice-sampling rate was 8 KHz, a signal of 1.536 Mbps (24 × 64 Kbps) was produced plus the one framing bit for an 8 KHz framing channel, which equals 1.544 Mbps. The framing channel is broken down as 12 m bits (4 Kbps for out-of-band channel management), 6 c bits (2 Kbps for CRCC), and 6 Fe bits (2 Kbps for framing). In South America, Mexico, and Europe the equivalent WAN channel is called E1 and has a data rate of 2.048 Mbps.

T3 Carrier

There are various hierarchies of T carrier, not all of which find current use in the United States. T3 does, however. T3, which is also an international standard known as DS3, consists of twenty-eight T1 lines (actually according to the international hierarchy, it consists of seven T2 channels, each composed of two T1C channels, which in turn are made up of two each T1s. T3 is an option for large corporations and government organizations since its cost—exceeding $100 a mile per month—is quite high.

Digital Data Service (DDS)

The two DDS digital signaling channels—a digital service unit (DSU) and its corresponding channel service unit (CSU)—transmit up to 56 Kbps (actually a multiplexed DS0), offering

digital transmission without the need for modems. These two units perform the services analogous to a modem on an analog line. The previous name for this service (as offered by AT&T) was Digital Dataphone Service. DDS is a leased-line (not switched) service that offers data rates from 2,400 to 56,000 bps.

Fractional T1

Fractional T1 is a line that consists of one or more DS0 channels. Multiplexers are required to put the information on a T1 line. Various fractional T1 combinations provide different amounts of bandwidth:

DDS	1 DS0	64 Kbps
H0	6 DS0	384 Kbps
H11	24 DS0	1.536 Mbps (T1 without framing)
H12	30 DS0	1.920 Mbps

Actually, most fractional T1 (FT1) runs at 384 Kbps (6 DS0), 512 Kbps (8 DS0), or 768 Kbps (12 DS0).

Fiber Distributed Data Interface (FDDI)

The original high-speed data bus, fiber distributed data interface (FDDI) is a fiber-optic token-passing ring. It may use single- or multimode fiber. Actually, FDDI consists of a dual fiber ring (primary and secondary) where information may circulate in opposite directions. More than one frame will be on the line at a time due to FDDI's high speed (this is the same method used with the 16 Mbps token ring). This high speed is the result of the Early Token Release (ETR), in which the transceiver is usually receiving its own signal before it finishes transmission and marks the poll bit as a token on the fly. There is always data on the line.

FDDI allows up to one thousand connections (not nodes). A node with both a primary and a secondary ring connection will count as two connections. FDDI does allow a 100-kilometer network span with up to 2 kilometers between any two nodes. (It can actually extend much farther—up to twenty times farther—if single-mode rather than multi-mode cable is used.)

FDDI has been modified from its original packet-switching format to be able to handle (if necessary) circuit switching for voice and video transmission. This is FDDI-II. FDDI requires 125 Megabaud media for 100 Mbps transmission because it uses the 4B/5B method of encoding, where 4 bits are transmitted as 5 bits. This represents an efficiency of 80 percent. (Conversely, 100 Mbps Ethernet requires a 200 Megabaud media. Two cycles of clock per bit time make for a zero and one cycle of clock for a 1, which results in a 50 percent efficiency of media capacity.) The encoding of 4 bit (Hex) groups by the five bits is such that for data no combinations of 5-bit patterns will have more than three consecutive zeros.

Metropolitan Area Network (MAN)

The Metropolitan Area Network is described in IEEE 802.6 (there are other MANs, but this is the only one we will describe). Installed in the United States usually in governmental centers

(Washington D.C.; Tallahassee, Florida; and Sacramento, California), MANs generally use ATM and a methodology called Distributed Queue Dual Bus DQDB. They are currently operated at 155 Mbps, with accommodation for SMDS with speeds up to 600 Mbps when Switched Multi-Megabit Data Service (SMDS) is standardized. The typical industrial user will probably not be concerned with a MAN, and if he or she were, it would be through a DSU-like device and at the data rate of the LAN (or near it).

Asynchronous Transfer Mode (ATM)

Asynchronous transfer mode (ATM) is also known as cell relay. It is primarily a fiberoptic transmission system and is highly suitable for that medium as the very small transmission packet called a "cell" requires it to use hardware routing, switching, and error detection. Data is transmitted in fifty-three octet packets (cells), with data taking up forty-eight octets while the remaining five octets are used for header information. ATM was originally scheduled to have a 155 Mbps data rate, although there are now different rates to suit differing applications, from 25 Mbps (ATM25) through the 600 Mbps that is envisioned running over the Metropolitan Area Network (IEEE 802.6). ATM is presently the technology of choice for carriers and handles voice, video, and data over WANs or LANs with a specified quality of service (QOS). However, high costs and slowness in developing standards have impeded the wide adoption of ATMs by anyone but wide area carriers.

Technically, for LAN users ATM LAN emulation allows the ATM connection to appear to be a token ring or Ethernet connection. The Ethernet (or token ring) packet is then encapsulated into the cell structure of ATM. When it appears at the other end of the connection it is appropriately reassembled into an Ethernet (or token ring) packet. ATM is probably more of a technique for carriers than a LAN methodology, particularly now that 1,000 Mbps Ethernet is in the process of being standardized. It was widely speculated that ATM would be the network of the future (particularly where deterministic services are required), yet this was when 10 Mbps Ethernet was standard. 100 Mbps Ethernet was two to four times as fast as Desktop ATM and was only a fraction of the cost. Even 1 Gbps Switched Ethernet is deterministic and has a cost structure lower than ATM. And while many network topologies and technologies were crowned successors to TCP/IP over Ethernet, 10 Gbps Ethernet is here. Moreover, 40 Gbps is on the horizon and, along with IPV6, eliminates complaints about QOS. Although ATM is the primary technology used in B-ISDN, SMDS, and the Metropolitan Area Network (MAN) and offered by other carriers providing WAN services, 10 Gbps Ethernet over fiberoptic lines though, is becoming a serious, viable contender in the long-haul business.

Synchronous Optical Network (SONET)

A SONET network can be configured in a number of topologies but is most often used as a dual counter-rotating ring topology. It essentially encapsulates whatever data appears at the interface. SONET has four layers that are roughly analogous to the OSI model. Its minimum data rate is 51.84 Mbps, and its maximum (at the moment) is 2.488 Gbps. Table 7-2 outlines the data rates available (all rounded off except for OC-1).

SONET networks have been proposed as the "fiber-to-the-curb" infrastructure in which copper will carry the transmission into individual residences—something like having a central office in every neighborhood. Though it is nice to know that these technologies exist and are being pursued by carrier companies; for those in the industrial field that information is all that is *needed* since almost any and all expansion beyond what a carrier company now uses is "proposed."

Optical Carrier Level	Transmission Rate Mbps
OC-1	51.84
OC-3	155
OC-9	467
OC-12	622
OC-18	933
OC-24	1,244
OC-36	1,866
OC-48	2,488
OC-192	10,000
OC-256	13,271

Table 7-2. Optical Carrier Data Rates

The Answer: Digital Subscriber Line (DSL)

Using the same copper pairs that bring the venerable voice-grade wireline to your residence or business, digital subscriber line technology, (x)DSL, has gained wide acceptance. It is considered the broadband technology because it can deliver more than one service at a time (it does not consist of a baseband signal). There are many varieties of DSL. Each requires that various modifications be made to the central office wiring and perhaps to the premise's wiring (eliminating impedance-matching devices that were designed for voice frequencies, leaving just the bare copper). It should be noted that DSL as offered is not just a connection medium but must be considered as a connection to the Internet. ISP services are offered in the price and connectivity is complete with one or more IP. One should ascertain that VPN or other security techniques are used if DSL lines are to be considered.

Among the many kinds of DSL are these:

ADSL – asymmetric DSL
HDSL – high-bit-rate DSL
RADSL – rate adaptive DSL
SDSL – symmetric DSL
VDSL – very-high-bit-rate DSL

Here we will only discuss the most often encountered type: ADSL. In ADSL, the upload data rate is generally less than the download data rate. Depending on which type of ADSL you use, the upload data rates may vary from 64 Kbps to 1.5 Mbps and the download data rates from 500 Kbps to 6 Mbps. Typically, these rates are tariffed quite reasonably, with residential service with an Internet service provider (ISP) costing about $19 - $24 a month (circa May 2007). This will allow you (typically) one IP address. Businesses, on the other hand, are generally given a more symmetrical rate (that is, upload and download are closer together in frequency) and several IP addresses for a business rate of about $90 (circa May 2007) a month. Either service includes the capability of carrying on a telephone conversation at the same time that data is being utilized on the DSL lines. This makes an attractive package: trading business telephone line costs for not only a business phone line but also high-speed Internet and ISP access for the same rate.

DSL lines are not totally digital in that they use modulation of sorts: either CAP (Carrierless amplitude and phase modulation—analogous to the techniques used by the cable TV system) or DMT (Discrete Multi-Tone—a proven technique for placing high data rates on lines of limited bandwidth). Originally used in wireline modems, DMT uses many tones, each of which is modulated with a part of the data. As a result, the decision rate for the media is quite low. One DMT scheme has the data divided into 256 channels, each with 4 KHz capacity. Each data segment is assigned a unique ID and is spaced all over the allocated bandwidth. At the receive end the segments are reassembled (by ID), and the packet is passed upward.

At some point in the near future, everyone could have a DSL line to their business, to their residence, and so on. There will be challenges along the way, however. DSL as supplied by the telephone operating companies use the telcos' installed copper voice pairs. The data rate varies inversely with the distance from the central office, and once you are past 18,000 feet or so DSL doesn't work very well at all without the use of repeaters. And to make matters worse, not all central offices support DSL. However, seeing the competitive marketplace, telephone operating companies are moving at an extraordinary speed (for them) to achieve wide coverage for DSL. If you aren't yet near a central office equipment that supports DSL, you may obtain DSL like speeds over radio links or even (rather reasonably) by up/down link satellite.

Cable Modems

Another Internet connectivity medium, though not deployed in a large number of industries is the cable modem. The total number of cable modems in use roughly equals the number of DSL customers (May 2006) although with the improved roll out by telcos, the number of DSL lines in use may significantly outnumber cable customers (depending upon whose marketing statistics you wish to accept). Providing (typically) a 512 Kbps to 6 Mbps data rate to customers when the cable segment is not loaded, data rates drop as loading goes up because all customers are using the same shared media (just like a real shared media network). As with the telcos, cable providers offer different rates and tariffs depending on the speed and service

you require. Cable modems have brought one key benefit: they gave telcos the impetus to implement DSL. Unfortunately, neither the cable modems nor DSL do rural and far suburban customers much good. These users are at present either stuck with 33.6 Kbps analog modems or must use a satellite system with its correspondingly increased costs.

In the next section we introduce some solutions for these customers. Several wireless schemes are proposed even for remote rural customers, but the direct satellite link is already here (typically costing $300 to install and near $80 monthly, circa 2007), and with the exception of occasional weather-related outages satellite can supply DSL data rates (or better).

WAN for the Mobile and Outer Lands

If there is to be a high-speed data service (more than 56 Kbps) to outlying areas in the future it will have to be wireless, which is a form of radio. One should be careful of the routing end points when using wireless; some are confined to point to point, others connect to the Internet. Routing would depend on the service or services desired. One form of wireless is the Universal Mobile Telecommunications Service (UMTS), which is a mobile technology that has a data rate of up to 2 Mbps. Then there are the other mobile technologies, such as General Packet Radio System, which offers a data rate of 56 to 114 Kbps, and the Enhanced Data GSM Environment, which provides a data rate of 384 Kbps. At one time, the most promising wireless technology for fixed (not mobile) customers was Local Multi-point Distribution System (LMDS), which had a data rate of 155 Mbps. It used technology similar to cell phones in that each transmitter covers a "cell," typically 3 to 5 kilometers in diameter, but the client is fixed and does not move. The technology is now used by IEEE 802.16(fixed) and 16e (mobile) under the name WiMax. This is certainly a promising technology for rural and perhaps urban customers.

At the moment, the only hope for rural areas that want high-speed access is by going to satellite. Because at this time (2007), most of the mobile services are cell based, and thus require a multitude of antennas and low-power repeaters. This does not appear to be a feasible or cost-effective way to deliver services to rural or isolated areas (unless they are next to a major highway). Residential users of satellite television, on the other hand, can get a hybrid service that receives data via the satellite at DSL rates, but must use analog telephone modems to upload data. Although some satellite suppliers will provide true up- and downlink satellite service (for a price), and provides connection to the Internet, it is the only high speed bidirectional option for many rural subscribers. An example is DirecPC, which is a satellite-delivered service with a data rate of 400 Kbps. Again remember this is an Internet connection, although the technologies could be employed otherwise as in a private network. Anything other than normal generally has an astronomical rise in pricing.

Industrial users might wonder why they need to even consider a wireless WAN. One reason is not because wireless WAN can reach isolated areas—in reality, few plants are built (other than generating facilities) in isolation, and transportation infrastructure is needed for such plants to operate. The main reason industrial users would consider wireless is that nearly

half the cost of any large networked installation is in the wiring (both initial and life-cycle costs). The benefits of wireless can be great, although the WAN application isn't needed (except for interplant communications). In the plant area, the technology developed for wireless may filter down to the LAN applications. Having unbound nodes allows a great deal of flexibility, but also comes with a host of problems, not the least of which is many walls made of materials that are highly absorbent of radio frequencies, large amounts of electrical noise, and many points for multi-path reception and thus the resulting interference. Yet spread-spectrum technology (particularly the frequency-hopping type) overcomes many of these problems. See our earlier discussion of wireless in chapter 3.

Table 7-3 illustrates some of the typical data rates for WAN services currently available.

Name	Media	Speed
Plain old telephone service (POTS)	Unshielded Twisted Pair (UTP)	3 KHz/up to 56 Kbps
Mobile Telephone (GSM)	Radiated (RF)	9.6 to 14.4 Kbps
Frame Relay DDS	Copper/fiber optic	56 Kbps
DS0 (basic unit)	All	64 Kbps
General Packet Switched Radio (GPRS)	Radiated (RF)	56 Kbps to 114 Kbps
Integrated Services Digital Network (ISDN) BRI Rate	Unshielded twisted pair	64 Kbps or 128 Kbps
Integrated Services Digital Network ISDN PRI Rate	Copper/fiber optic	1.544 Mbps or 2.048 Mbps
IDSL	Unshielded twisted pair	128 Kbps
AppleTalk	Unshielded twisted pair	230.4 Kbps
DirecPC	Satellite RF	400 Kbps
Frame Relay	Unshielded twisted pair or coaxial cable	64 Kbps to 1.544 Mbps
T-1 (DS1)	Copper/fiber optic	1.544 Mbps
E-1	Copper/fiber optic	2.048 Mbps
(x)DSL	Unshielded twisted pair	512 Kbps to 6 Mbps
Cable Modem (subscriber)	Coaxial cable	512 Kbps to 6 Mbps
Cable Modem (to ISP from head end)	Coaxial cable	52 Mbps
T-3 (DS3)	Coaxial cable	44.736 Mbps
FDDI	Fiber-optic cable	100 Mbps
OC-1 (basic unit)	Fiber-optic cable	51.84 Mbps
OC-3	Fiber-optic cable	155.52 Mbps
MAN	Fiber-optic cable	155 Mbps to 622.08 Mbps
SONET	Fiber-optic cable	51.84 Mbps to 2.488 Gbps

Table 7-3. Selected WAN Data

Summary: WAN

Our brief discussion of wide area networks has hopefully familiarized you with some of the outside services and terminologies. Many wide area networks are in service today—many of them packet-switched, digital networks that use different protocols. Though some may have industrial applications, many are better suited to the data traffic required for database transfer and point-of-sale (POS) terminals. The data rates that are cost-effective today are relatively low (compared to a LAN), but system evolution will change that. There are ways to make standard long-distance media (more than 50 meters) accept higher data rates, and several standard networks do so. Packet switching is a way to take a large continuous stream of binary data (messages), break them up into smaller frames or packets, and transmit these packets through the public switched network at whatever speed the link will support. X.25 is the primary standard that covers packet switching for open systems, but it is slowly losing out to frame relay. Connectionless frame relay operates at much greater speeds and efficiency than X.25, provided that errors and congestion are not formidable. ISDN is a concept that could have combined the digital phone along with digital information, but its window of opportunity has passed. Now the competition is between ATM, DSL, and 1000/10000 Megabit Ethernet, as well as the ever-present threat of cable modems.

Though little in this chapter may appear to be directly related to process control, it certainly is related to industrial data communications, in that all of the concepts and equipment now used in industry began use in general data communications. Many of the concepts presented here are already working themselves into the industrial area. And with the advent of corporate "intranets" spread over a large geographical area, industry's adoption of the WAN seems a foregone conclusion.

Bibliography

Note that Internet links may change.

About.com. "Wireless and Networking." http://www.compnetworking.about.com/od/dslvs-cablemodem/a/dslc

"DSL." http://www.technical.philex.com/networks/sharing/dsl.htm

Feibel, Werner. Encyclopedia of Networking. 2d ed. San Francisco, CA: The Network Press, 1996.

Fortune. "Broadband." http://www.fortune.com/fortune/sections/broadband.

Langley G. Telephony's Dictionary. 2d ed. Chicago: Telephony Publishing Co., 1986.

Stallings, William. ISDN: An Introduction. New York: Macmillan Inc., 1989.

Webopedia [online dictionary and search engine for computers and Internet technology. http://webopedia.internet.com

Whatis.com [online IT encyclopedia and learning center]. http://www.whatis.com/thespeed.htm.

8
Internetworking

Internetworking is really the subject of this book—that is, the seamless transfer of information throughout an enterprise even if that enterprise has plants and offices throughout the world. The application of internetworking could rightly be called systems integration—the bridges, routers, and gateways *portion of the text*. Myriad are the ways of internetworking. They are generally based on standards, however, because it is not only industrial applications that have confusion at Layer 7. Some of the information on internetworking in this chapter reviews previously discussed information to ensure that key concepts are understood.

Layer 2: Internetworking Equipment

A Layer 2 device is so named because it reads the data link information and uses it to perform some action. A bridge is a Layer 2 device and so is a switching "hub." Both read the packet's Layer 2 destination address (MAC). Both could (but don't have to) identify protocol (for 802.X packets) for the purpose of filtering caused by reading the type/length octets. Figure 8-1 illustrates a Layer 2 device. First, a definition of some of the internetworking devices (or review depending upon your point of view) is necessary.

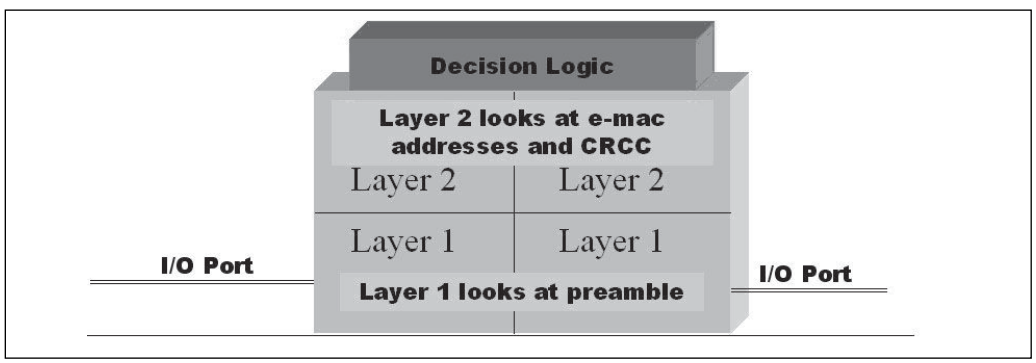


Figure 8-1. Block Diagram of a Layer 2 Device

Switch

A switch contains circuitry that turns each connection between nodes into a virtual mini-network, thus simulating a bridged LAN with multiple nodes. A switch has a processor(s) and an electronic crossbar that replaces the hub bus. The processor checks the address and uses the interconnect to transfer the frame to the correct port. Modern Ethernet switches can auto-detect speeds and perform conversions for 10-100-1000 Mbps Ethernet.

Bridge

A *bridge* is a device that contains two sets of Layer 1 and 2 functions and connects network segments that have the same type. It reads the Layer 2 addresses (source and destination e-macs), and develops or matches (depending on whether it is transmitting or receiving a frame) the crcc. The bridge uses an internal table called a "bridge table" to determine which side of the bridge that devices reside. A switch may be considered a multiport bridge doing all of the same actions but for (typically) more than three ports.

Two methods are used to cut the number of collisions on a many-node shared segment: (1) increasing the speed (upgrade from 10 Mbps to 100 Mbps) or (2) as a bridge does not pass on collisions, the use of a bridge to break up the collision domain into smaller domains so contention occurs only with the nodes in one particular collision domain. Collisions are a MAC Physical layer procedure and are not passed through the LLC (Layer 2) and hence, not by a bridge.

Types of Bridges

Several different devices are called bridges; some actually are bridges, and some aren't. Generally, bridges are classified as:

- Static
- Learning
- Transparent
- Translation
- Source route

Let's start with a basic bridge: the static bridge (see figure 8-2).

Static Bridge

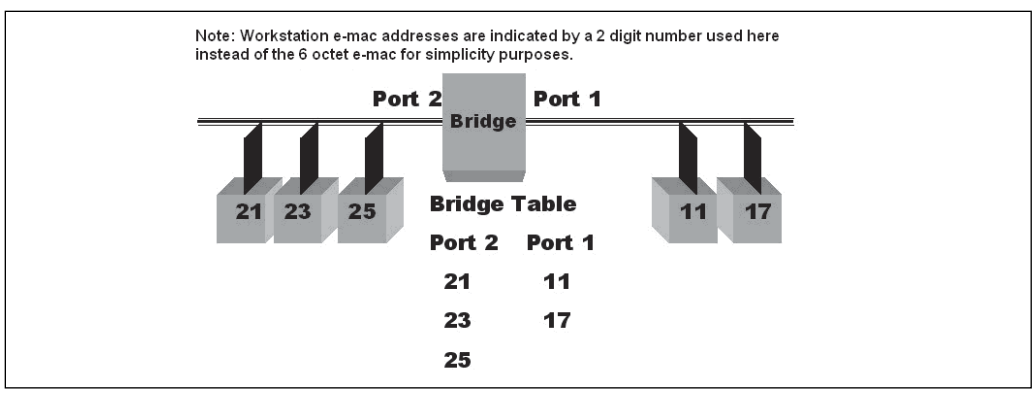


Figure 8-2. Static Bridge

In a static bridge, you had to physically enter the workstation e-mac addresses into the bridge table. The purpose of this was to ensure that the bridge would *forward* packets across the bridge when necessary (example: workstation 11 sends to workstation 23) and

block packets that don't need to be forwarded (example: workstation 21 sends to workstation 25). Physically entering the workstation e-mac addresses into the bridge table reduces contention across the whole network, leaving each segment as a contention (collision) domain.

Static Bridge Problems

A static bridge is not dynamic; changes are not reflected unless the administrator manually enters them. In a large network, manually computing a bridge table is time consuming and always resplendent with error.

Learning Bridge

The learning bridge is the contemporary model. When first activated it listens to the network talking. Since the source address is part of the Layer 2 frame, it quickly builds its own bridge table. If you move a workstation from one side of the bridge to the other it may take a while (sixty seconds or so), but it will soon determine which side you are now on and forward or block accordingly. When it is first activated the workstation takes thirty to sixty seconds to determine the bridge table. It is dynamic in that changes (workstations coming on or off of the network segment) are reflected within a short time period. Learning bridges require no intervention on the administrator's part. All modern bridges are learning bridges.

When the bridge is initialized there will be a period in which only some addresses will be in the table. If a workstation once was on the network and then went off (for a short time before the bridge declared it inactive) then the address table might be wrong for that period of time.

Bridging Actions/Functions

All bridges perform the following functions:

- filtering
- forwarding
- blocking

Forwarding and blocking have actually been described. If a device is on other than the same segment from which it originated it is forwarded by the bridge to its residual segment. If it is on the same segment as it originated, then it is blocked from traversing the bridge. We will describe filtering in the following section.

Filtering

A bridge operates in the "promiscuous" mode. In other words, it reads every packet on each segment, not just the ones addressed to it. Filtering means the bridge matches a packet to its bridge table. This produces one of three results: the match is on the same port, the match is on a different port, or there is no match. Let's examine each of these results in turn.

Same Port Match

Referring to figure 8-2, note that if the bridge receives a packet from workstation 21 for workstation 23, it looks in the table and it has a match—on the same port! Therefore, it blocks the packet so it does not cross the bridge. It doesn't need to, however, and the bridge provides no action at all on this packet other then denying it travel across the bridge and checking the CRCC.

Different Port Match

Referring to figure 8-2, note that if the bridge receives a packet from workstation 21 for workstation 17 it finds a match in the table—on a different port! The bridge therefore will *forward* the packet across the bridge.

Flooding: No Match Found

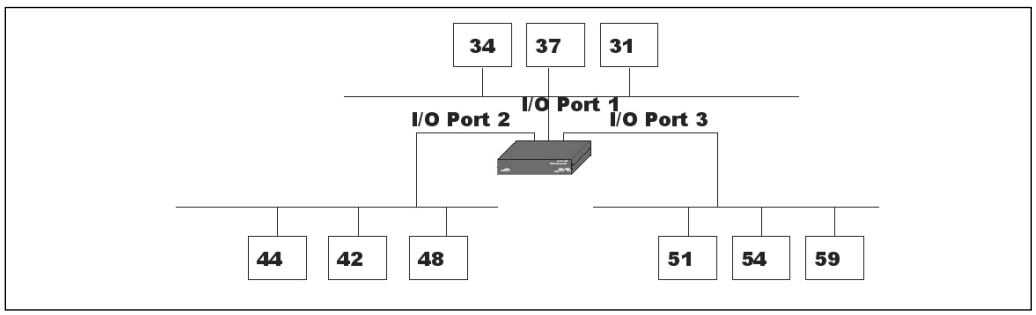


Figure 8-3. Flooding

Referring to figure 8-3, note that if the bridge receives a packet with a destination address that cannot be found in the bridge table (example: workstation 34 wants to send to workstation 79), the bridge *floods* all ports (2 and 3) except the packet-receiving port (1). Though a two-port bridge would merely forward to the other port, bridges with three or more ports would "flood" this packet to all ports in hopes of obtaining an answer from one of them. This could conceivably cause a "duplicate" packet problem. Ethernet does not allow for duplicate packets, nor do most Layer 2 protocols. The duplicate packet problem could arise if there are loops in the network.

The Loop Problem

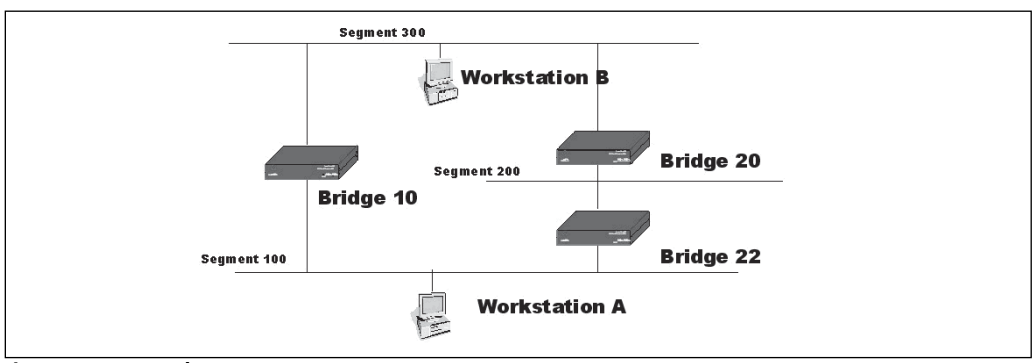


Figure 8-4. Network Loop

Referring to figure 8-4, you will notice that packets between Workstation A and B can arrive by two different routes. This dual pathway is called a "loop." It is not desirable, however, because duplicate packets cause problems, and Layer 2 has no way to correct it. Particularly on an Ethernet network, *packets must have one and only one path* between any two devices on the network. Ethernet in particular uses transparent bridging, where transparent means the user takes no actions whatsoever and the bridge must ensure correct connectivity.

Transparent Bridging

A transparent bridge is one in which the loop problem is taken care of without intervention by the user or administrator. It falls solely to the bridge to detect and eliminate network loops—and therefore actions taken to eliminate loops are transparent to the end user. The method used to accomplish loop elimination in a transparent bridge is to implement the Spanning Tree Algorithm (STA).

Spanning Tree Algorithm (IEEE 802.1d)

The Spanning Tree Algorithm is an IEEE 802.1d standardized method (and therefore the only one used) that enables bridges to detect and eliminate network loops (see figure 8.5). It uses configuration messages (which add more traffic to the network) between bridges to determine a single path between segments. It does this by developing a tree structure—one root, many branches. STA determines a root bridge and then blocks certain bridge I/O ports to open the loops.

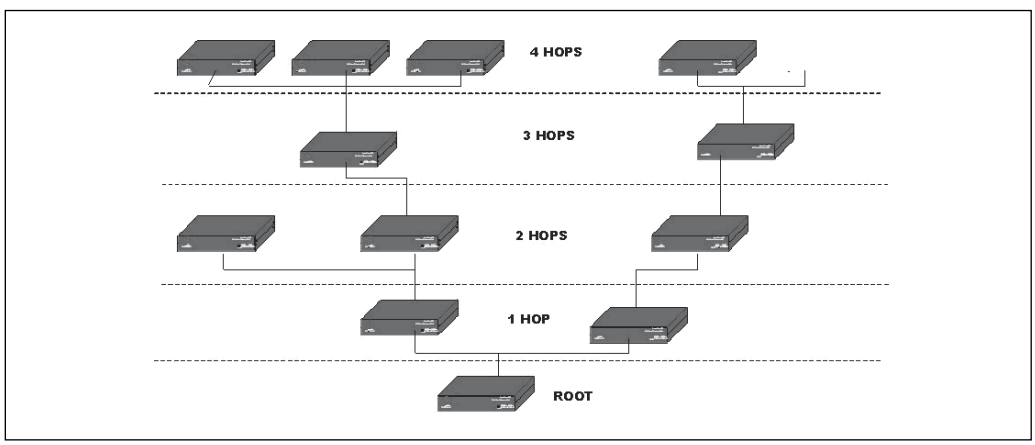


Figure 8-5. Spanning Tree Diagram

The logical tree design of the spanning tree algorithm is illustrated in figure 8-5. There is only one path to every location. Bridges are assigned an ID. The algorithm *to create the tree and locate devices in the tree structure* uses this ID. For our purposes, a configuration message should have three parts:

- root ID
- number of hops
- bridge ID

The tree is set up using spanning tree configuration messages. When a bridge is initialized it assumes that it is the "root" and broadcasts a configuration message. To express this, we will use a symbolized (in other words, not actual) message that has the format: root id, number of hops, this bridge id. In figure 8-6 the root is bridge 2. Its configuration message would be 0002, 00, 0002.

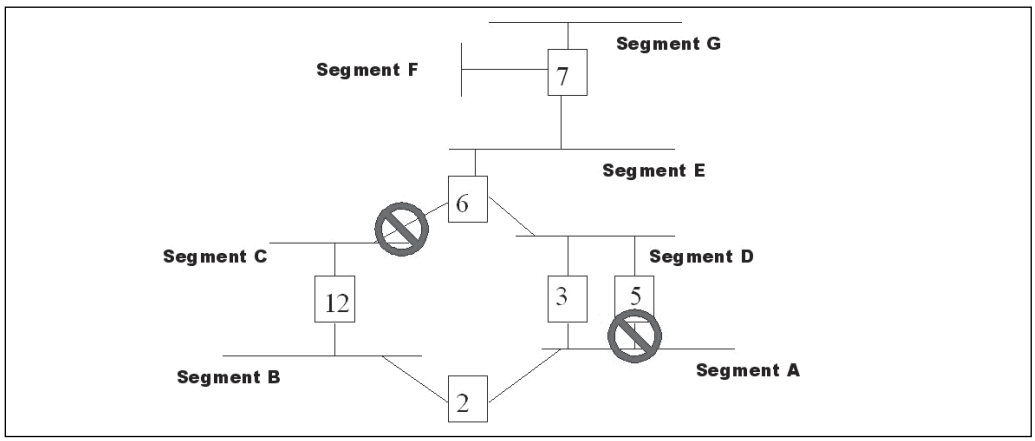


Figure 8-6. STA Example

When bridge 5 comes on line, it would transmit the message 0005,00,0005, assuming it was the bridge. However, bridge 0002 has a lower ID, so when bridge 0005 hears bridge 0002, it will drop its claim as root and acknowledge 0002 as root.

Once the root is established, loop detection starts. Both bridges 0003 and 0005 are one hop away from bridge 0002. Therefore since one Ethernet hop is a cost of 10, and both bridges have one hop from 0002, they each have a hop cost of 10. So their messages would be 0002, 10, 0003 and 0002, 10, 0005. Since both bridge 0003 and 0005 provide a path to segment D, a loop would be formed. In this case, the bridge with the highest ID (0005) would block its highest numbered port between the segments. Note that the same would apply to bridge 6. In this way, the spanning tree algorithm automatically prevents loops.

Summary: Spanning Tree

It is not necessary that you be able to compute a spanning tree algorithm. However, it is essential that you understand its basic concept—automatically preventing network loops. Crucial to your understanding of bridges or any Layer 2 device is that the lowest ID is the selector and an overview of the configuration message concept. All modern bridges (and most routers in bridge operation) support the spanning tree algorithm. Most industrial bridges and switches use a slightly different algorithm called the Fast Spanning Tree Algorithm. This algorithm does not need as much time to determine the root and tree upon startup or after loss of the current root. However, transparent bridging and the equipment implementing the bridge are not the only bridges in use. In most cases transparent bridging

only applies to Ethernet, and there are a host of other network protocols, particularly in industrial networking that must be interconnected on a Layer 1/Layer 2 basis.

Translating Bridges

A translating bridge passes (or bridges) packets between two different networks, such as 802.3 (Ethernet) and 802.5 (token ring). This is actually a data link gateway and not a bridge per se. A translating bridge must change between big endian (802.5) and little endian (802.3) transmission, different packet sizes (17K versus 1,518 octets) and different bridging techniques (source routing versus spanning tree). Though this may be accomplished using Layer 2 information (that's why it's a bridge), a translating bridge functions much more as a protocol converter. Source routing is a different concept than transparent bridging and requires an interface between the two bridging techniques: transparent and source routing.

Source Route Bridges

Used in token rings, a source route bridge places the Layer 2 packet routing and loop detection on the client rather than the bridge. In source routing, the station transmits a discovery packet that goes to all routes. The first returning packet (which has all the hops and bridge numbers it used along the way) determines that packet's routing for the rest of the message. Source routing is not as efficient as the spanning tree algorithm in that it depends on the client rather than the bridge. Token ring can use the spanning tree algorithm provided that all its bridges can do so too.

Remote Bridges

Many bridges, called remote bridges, have provision for wide area connectivity and enable bridging across the WAN. One concern with this remote bridging methodology is its lack of bandwidth in most WAN applications. Even a T1 line may be a bottleneck on a 10 Mbps LAN, let alone a 28.8 Kbps connection. If a WAN needs to be crossed, a bridge is not the answer, and a Layer 3 device should be considered.

Switches As Bridges

A switching hub reads the data link addresses, just as a bridge does, and determines which port to connect. A switching hub should be considered a multi-port bridge. For both switches and bridges there are some considerations one must take into account with complex connections. A multiple path for packets at the Data Link layer is not allowed, for example. Higher-end (higher-priced) switches will have provisions for bridging as well as switching and will contain the spanning tree algorithm. However, not all switches are created equal. It is essential that good wiring practices be followed (EIA 568 calls for just three levels of wiring) and a tree hierarchy be strictly implemented. Since we are dealing with industrial networks redundancy is relevant. Network redundancy is common in industrial systems; however, these are special cases and are not addressed in the general networking scheme. This is why *redundant industrial Ethernet systems or any other redun-*

dant system dedicated to industrial requirements are generally a vendor specific implementation. Even with redundancy only one copy of a packet can arrive at the receiver since Layer 2 has no way of correcting for multiple identical packets. For tying together multiple networks, or for organizing an industrial system into sub networks, a Layer 3 device is best employed.

Layer 3 Devices

To address Layer 3 devices properly we will discuss the following topics:

- Layer 3 packet information
- IP router actions
- router protocols
- advertising
- RIP
- Link State Protocols
- OSPF
- bridges versus routers
- multi-protocol routers
- hierarchical routing
- interdomain protocols
- VLANS

Now, is all this essential to an industrial networking person? Absolutely! As we stated at the outset of this chapter, industrial systems are becoming a part of larger systems. This includes having their packets go other places–not simply within the little island of automation that they have been restricted to in the past. Layer 3 is thus the heart of internetworking. It is where you find the network you want to address; it is where you find out how to route to the location you want.

Layer 3 Packet Information

In this section we will deal primarily with TCP/IP over Ethernet, the LAN standard. You will recall that the type/length octet determines the protocols to follow. If the two octets are fewer than 1500 decimal (the maximum number of information octets in an Ethernet), it is expected that the 802.2 control octets (2-3) follow. If the two octets are more than 1500 decimal, the frame is a non-802.2 frame, and the octets identify which protocol will follow. If the two octets are 2048 decimal (0000 1000 0000 0000 = 0800 hex or 2048 decimal) then the next information will be an IP header. And if TCP is the transport, then somewhere in the IP header this will be stated, and TCP octets will follow the IP header. Bits 12-20 (in a 20-octet IPV4 header) will contain the source and destination address. Of course, all this eats up some of the 1,500 octets reserved for data in this frame.

Layer 3 devices are protocol sensitive. A Layer 3 device must be designed to read the protocol (know what each bit represents) that it is handling. Since these protocols contain the addresses for different machines on different networks, they are essential to routing. A routing protocol is necessary.

Router Actions

For purposes of explanation, figure 8-7 uses a simplified notation of addressing. The decimal number by the workstation stands for the six-octet e-mac (Layer 2) address of the NIC. The network address (Layer 3) is the decimal number that identifies the network on which the workstation(s) reside.

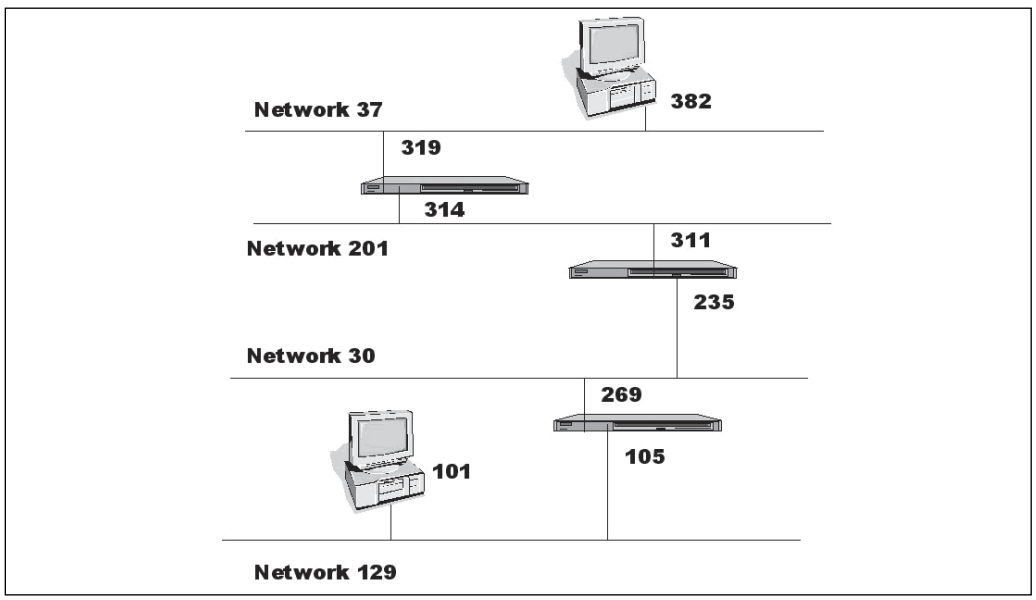


Figure 8-7. Simplified Router Addressing

When a router receives a packet that is destined for another network, it must create a route or path to the foreign network in the form of a list of router hops. Every time a router receives a packet and hands it off, it rewrites the data link address for the next router to which the packet will travel. Each router has to estimate on its own on which path each packet should be sent.

Figure 8-7 is a simplified explanation of how a packet goes from a workstation with a Layer 2 address of 101 to a workstation with a Layer 2 address of 382. As you can see from figure 8-8, we have made the address a combination of the network and e-mac. It is not this easy in real life. Using the Internet Protocol involves using the four decimal notations for addresses. Someone (either an lmhosts file or DNS) has to resolve network names to e-mac numbers. For our purposes, let us assume that this is accomplished without further discovery on our part. That station at node 101 knows the destination network address from previous communications.

DL Destination Address	DL Source Address	Type/Length	Network Source Address	Network Destination Address
105	101	2048	129-101	37-382

Figure 8-8. Simplified Frame Information

Note that the Data Link layer (DL in figure 8-8) is only aware of addresses on this network. If the e-mac it is looking for is not on this segment then it will plug in the e-mac of the default router, in our case 105. This frame (remember this is an abridged and shortened frame model for purposes of discussion) goes to the router at 105. Router 105 would look in its routing table and see if a path exists to Network 129. Since we will discuss discovery later, let us assume that the router's routing table has that path. The router will move the packet across the router and change the Layer 2 addresses to reflect the next router in line, as shown in figure 8-9.

DL Destination Address	DL Source Address	Type/Length	Network Source Address	Network Destination Address
235	269	2048	129-101	37-382

Figure 8-9. Second Step in Network Addressing

You will note that the Layer 3 addresses remain unchanged, but the Layer 2 addresses reflect moving the packet on the Network 30 segment. The router at 269 moves it across and transmits the packet on Network 201. See figure 8-10 for the Layer 2 addressing.

DL Destination Address	DL Source Address	Type/Length	Network Source Address	Network Destination Address
314	311	2048	129-101	37-382

Figure 8-10. Third Step in Network Addressing

The packet crosses this router, arrives at the destination network, and is transmitted from the router to node 382. Figure 8-11 shows the last Layer 2 addressing.

DL Destination Address	DL Source Address	Type/Length	Network Source Address	Network Destination Address
382	319	2048	129-101	37-382

Figure 8-11. Last Step in Network Addressing

Note that (for Ethernet routing) it is Layer 2 that moves the packets on a network segment, and it is the destination address that moves them to the final destination. If you were using TCP/IP, the addressing would be different. Figure 8-12 would be what you would look for as an example.

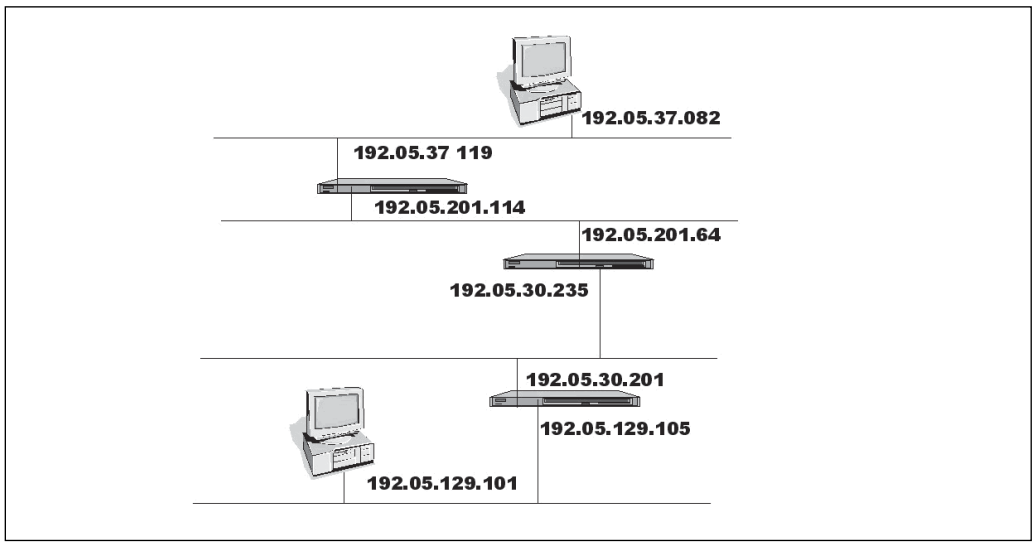


Figure 8-12. IP Routing

In figure 8-12, all the computers are on one Class B network, and the segments are separated by subnet masking. The addresses looked at would be the Layer 3 addresses. But how does the packet move on its own network with the Layer 2 frame? There must be a mechanism for equating a four-dot notational IP address with the e-mac for that device. Enter the Address Resolution Protocol (ARP). The ARP compares the destination IP address with every outbound IP Datagram to the ARP cache in the NIC card that will transmit the frame. If there is a matching entry then the e-mac address is obtained from cache and placed in the frame's destination address. If there is no match then the ARP broadcasts an ARP Request packet onto the local subnet (in our case, 129 if 101 is the one wishing to transmit). This packet asks the owner of the IP to reply with its e-mac. If the IP address does not lie on this segment, then the default router's e-mac is used, and the packet goes to the router. This would happen for each segment the packet went through until it arrived at its final destination. And this is why the default gateway (which is actually a router) address is so important on a network.

Subnet masks are used to take an address space like the Class B and apportion it as separate segments, each their own network. The subnet mask itself is a thirty-two-bit number that the receiving device uses to separate the Network ID from the Host ID. The easiest way to look at this is as follows: the Network ID is assigned 1s, and the Host ID is given zeros. It should be noted that subnet masking does not generate more addresses; it generates fewer from a given space. Each device on a logical subnet must have the appropriate subnet mask.

If you want to be able to browse (i.e., see the device in Windows' Network Neighborhood) you will either have to have a DNS or WINS server. If you lack either, you will need to install a "hosts" or "lmhosts" file. (Unix systems only require a "hosts" file.) This file relates the four-dot notation IP address with the Computer Name, an essential item for Windows messaging.

Two assumptions were made to make this packet move from one device to another: (1) that the originating node knew where it wanted to go, and (2) that the router knew the path. These assumptions are not always true, and therefore the reasons routers (like bridges) must learn. One of the first things they learn is which routers are their neighbors.

Routers will use an algorithm to dynamically decide where to send the packet next, based on a cost calculation. Other options such as least congested or fastest link time are available, but for WANs they tend to cost more both in terms of money and time. For LANs these options would cost more in response time and in number of hops to the destination.

Advertising

Computers on a segment know the e-mac of their router by "advertising." A means by which the router determines what addresses are available to it. The exact method of advertising varies with the protocol being used. Some have the end users and the routers broadcast advertisements; some have the router send out hello packets so the end user knows their e-mac. At this point, the router learns the e-mac from the end-user-transmitted frames.

Advertising Router Protocols

Advertising acquaints the router with the end users on its segment(s) but doesn't tell the router about other routers. Routers must share connectivity information with each other so they know how to get to any particular end user. Routers use routing protocols to build routing tables. Two major routing protocols are used: RIP and OSPF.

RIP

Routing Information Protocol (RIP) is a distance vector protocol. The protocol derives its name from the fact that it uses a single metric (measurement of path length): hops. When a router first comes on (using RIP) it sends out an update—who it is and whom it is connected to. RIP updates are sent every sixty seconds. The router listens to all the other routers that are sending updates. From these updates, the router builds a routing table. These tables are continually being modified and are broadcast with the update. In a system that has multiple routers it is easy to imagine how this information may be false for sixty seconds or so (sometimes much longer): in other words they tend to believe "gossip."

RIP Routing Tables

A simulated RIP routing table is illustrated in figure 8-13.

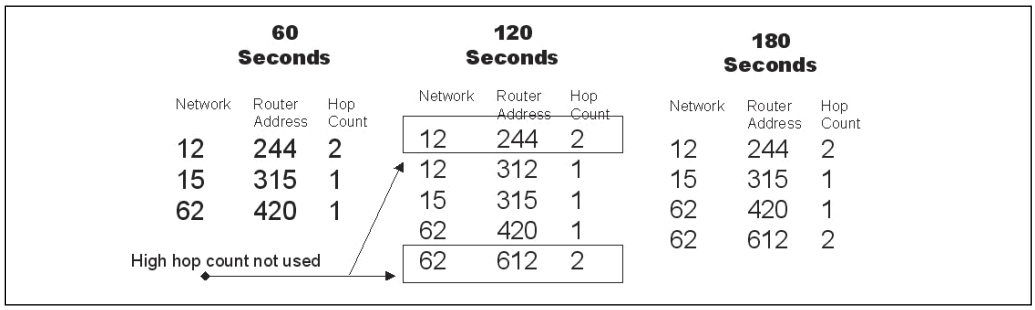


Figure 8-13. Simplified Routing Table

Note that the table has three entries at 60 seconds. At 120 seconds the higher hop entry would be erased from the table for both 12 and 62. This is because an RIP table will only indicate one (and only one) route, and this is the route with the shortest hop count (which may not be the highest bandwidth route). If Network12 address 312 becomes unavailable due to link or router problems, it will be at least 60 seconds before the Network 12 address 244 is reestablished.

RIP Problems

Routers using RIP transmit their tables periodically whether any changes have occurred or not. This wastes bandwidth. Convergence (as used with routers, not voice-over-IP) is a measure of how long it takes all the routers to be notified of a change. Because a router doesn't know where it is in relation to other routers (only how many hops there are between routers based on RIP) it believes things to be true which are not and may very well believe an incorrect hop count as well or as defined with technical elegance, it believes "gossip."

Gossip Explained

Refer to figure 8-14. If router 50 fails, router 10 will find out through the table broadcasts (in as much as five minutes). Router 10 still believes that routers 20 through 45 can connect to Network 5 because each has broadcast that it can. Because RIP only uses hop count, router 10 does not know the topology. The hop counter will count down to infinity for all packets routed to Network 5 (infinity is fifteen hops in RIP) before router 10 realizes that whichever router it utilized did not connect to Network 5. If router 10 were the only router trying to send to Network 5, it would have to try each router. This will take some time. Because the RIP routers believe everything they hear, it is called "gossip."

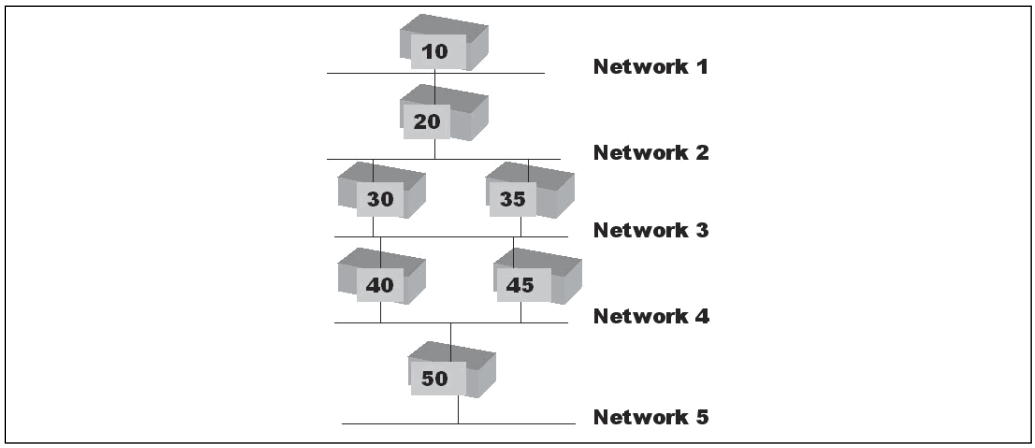


Figure 8-14. Gossip Network

Several methods have been worked out to compensate for these problems, but they generally cause as many problems as they cure. For example, *Hold Down* keeps the last news for a while before receiving any more. Similarly, in *Split Horizon* (poison path) the originating router can't receive information it sent out. It is beyond the scope of this text to go into these methods in detail, but an Internet search will yield any number of explanations for these two corrections to RIP gossip.

It takes a long time (relative to the speed-of-information rate) to propagate changes through the routers (convergence), and the information the router receives may or may not be accurate (gossip). Because of these faults (although we could make some corrections) inexpensive processing power has made available a different approach.

Link State Protocols

Link state protocols use more information than hop count, such as link speed, location, and the like. They develop a topology map in the router, and since they allow for multiple routing paths, they can provide load balancing. Link state protocols converge much faster than distance vector protocols. The protocol we will discuss is the OSPF (Open Shortest Path First).

OSPF

OSPF uses *Link State Flooding* to inform other routers which LANs this router is directly connected to (accurate information). OSPF then performs the *SPF Calculation*, which is the shortest (fastest) path to a distant network. Last, OSPF performs *Neighbor Discovery*, which locates routers that are directly attached to the same networks as this router. Let's consider each of these parts in turn.

Link State Flooding

Routers send out information concerning networks that they are directly connected to in the form of an advertisement. This is the Link State (link status) and includes link speed as well as where in the topology the routers are located. This information is distributed in a

Link State Packet (LSP), which has a sequence number and time stamp to prevent confusion. Routers collect this accurate information to build a topology map of the entire set of interconnected networks.

SPF Calculation

The SPF calculation determines the shortest path (in terms of link speed and number of hops) from this router to any other router that this router knows about. To ensure synchronization between the router databases, all routers broadcast at a set period, usually between 2.5 and 4 hours. This schedule will resynchronize and rebuild the system topology map in each router.

Neighbor Discovery

Routers use these Link State Protocols to discover routers that are directly connected to the same networks to which they are connected. Routers are organized into logical areas, called areas (that figures), autonomous systems, or domains (not to be confused with NT domains). And with the hierarchy will come different names for routers with different functions in the hierarchy, one such being the "designated" router.

Designated Routers

Domains can contain multiple autonomous systems (AS) in an OSI view. One router is designated to be the reference router to whom all other routers synchronize their topology maps. This reference router is called the designated router. The path from a router to the designated router is called an adjacency. More on hierarchies in a moment.

Bridges versus Routers

Should you use a bridge or a router? Figure 8-15 provides a quick summary of the advantages and disadvantages of each.

	Bridge	Router
Flat topology	●	
Segmentation		●
Multi-path		●
Reliability (data)		●
Inexpensive	●	●
Fast (data)	●	●
Protocol Dependent		●

Figure 8-15. Bridge/Router Comparison

The major advantage of using routers is segmentation of the network: breaking it up into organizational units. With segmentation, troubles in one network normally cannot affect other networks in a routed system. This allows a quarantine to be set up if needed. Routers add reliability by allowing fast convergence (after link problems) and by providing alternative paths for data.

Multiple Protocols

When you are using a multi-protocol router (remember, routers are protocol dependent) you can keep protocol information in two ways:

1. Separate databases for each protocol ("ships in the night"),
2. One database shared between protocols (integrated).

Let's discuss these next.

Ships in the Night

Keeping separate databases for each protocol ensures that problems that occur in one protocol do not affect the other protocol. The users are effectively in two different networks, and the synchronization and upgrading of link information occur separately. In effect, you have two different routers.

Integrated

Using just one database for both protocols means that problems affect both protocols. Protocol updates are done simultaneously, which saves bandwidth. Using a shared database means you have only one set of users and one set of configuration settings, a fact not to be taken lightly if you have to administer many routers.

Routing Topologies

Most routers are found in a hierarchical topology. Hierarchies are established to help organize routers. As such, areas communicate with other areas using the *designated router* (Level 1). The routing rules are as follows:

- Level 2 routers talk to Level 2 routers at the backbone level.
- Level 2 routers talk to the designated Level 1 router at the area level.
- Level 1 routers can only talk to Level 2 routers outside their area.
- Level 1 designated routers talk only to the routers in their area.

Figure 8-16 is an example of hierarchical routing.

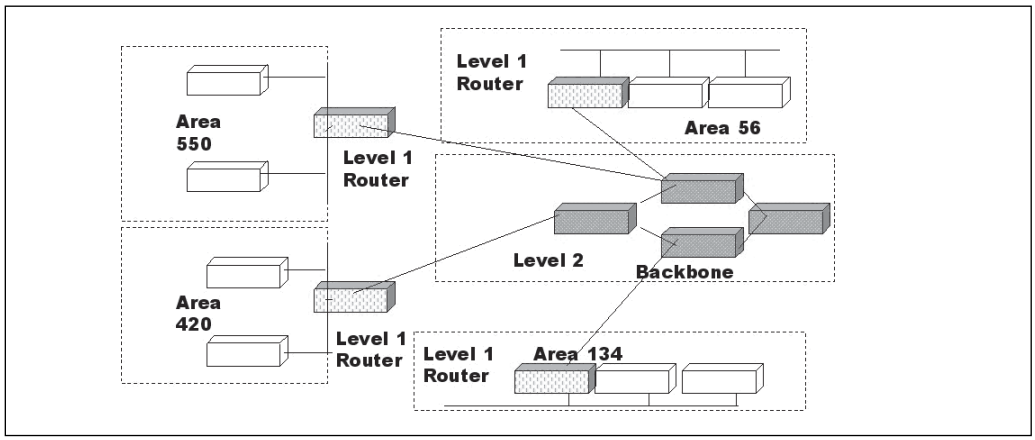


Figure 8-16. Hierarchical Router Topology

Router Physical Connections

Routers may have one-to-one, one-to-many, or many-to-one I/O. Typically, a departmental multi-protocol router will have an Ethernet connection (for the LAN side) and a WAN connection on the routed side (FRAD, T1 CSU/DSU, or EIA 232). Sometimes hubs or switches are incorporated on the LAN side. When this is done the routers are normally referred to as brouters, departmental routers, or, in the current lexicon, a "Layer 3 switch." These devices bridge on the LAN side, and when a packet is off network, the router forwards the packet.

VLANs

If a traditional IP router is used to separate network areas, then the network is divided into subnetworks. If a Layer 3 switch (the brouter or Divisional Router) is used, you can divide the network into a number of Virtual Local Area Networks (VLANs). In either case, the router or Layer 3 switch is the main locus for all the network traffic.

Routing switches have the ability to create virtual LANs (VLANs) in which the switch allows defined devices on different ports to act as if they are on the same LAN segment. VLANs group arbitrary collections of end nodes on multiple LAN segments into separate domains. The packets between VLAN nodes are switched, and the packets between VLANs are routed. This is useful for zoning or dividing the plant floor organizationally. Nodes may be included in both VLANs; however, the separate VLANs do not communicate with each other. This is identical to a firewall router DMZ zone.

When two devices are defined as being on the same subnet or VLAN, the switch passes through messages without doing any filtering, just as if the devices were on the same physical segment. However, if two devices are not on the same VLAN, then the switch runs the message through its filtering software, passing or blocking the message as appropriate.

Remember that the router or switch can only filter traffic that passes through it. It cannot separate two devices that are physically wired to the same segment. Thus, if it is important to filter traffic between two different groups of devices, make certain that they are attached to different switch ports. The best way to ensure this is to connect each device to its own switch port, creating a fully switched network. Figure 8-17 illustrates a typical industrial VLAN.

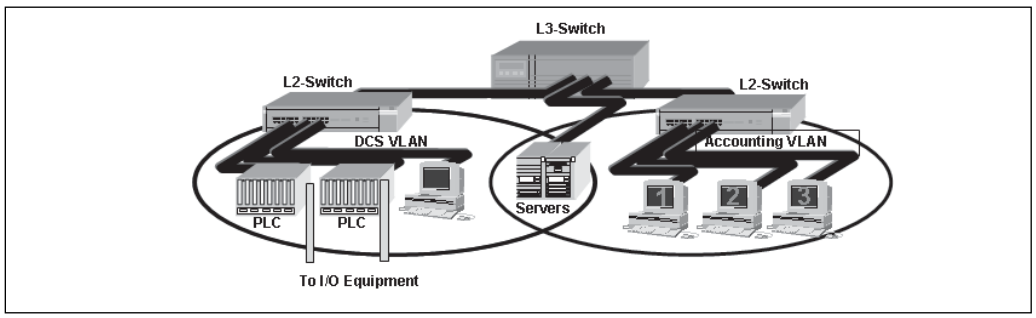


Figure 8-17. Fully Switched VLAN Network

Managed Switches

The features and configuration of managed switches vary with manufacturers and model by model. The configuration interface used for managing the switch is generally some form of Web interface, although there will be limited hardware configuration elements (notably a reset switch). Smart (or intelligent) switches are managed switches that have a limited user-changeable set of features.

The configurable features most likely found on a modern managed switch are:

- port availability
- port priority
- spanning tree algorithm/fast spanning tree algorithm
- SNMP (Simple Network Management Protocol)
- link mode port
- link aggregation
- VLAN settings

Currently, Layer 3 switch vendors are working to make the managed switch better so it not only examines packets and delivers them to the right port, but also examines whole data streams and takes some desired action on them. This functionality is being added in order to improve security and performance. Today, these different network goals require several network devices to accomplish what the new Layer 3 switches will be able to do by themselves. This will reduce equipment count and complexity and increase performance.

One of the problems with the present multiple-device approach is that infrastructure connectivity devices tend to be purchased by different groups within the organization. For example, IT purchases for load balancing, Security (usually but not always part of IT) for fire-

walls and IDS, and Engineering for VPN, VLAN, and gateway equipments. If two (or more) devices end up conflicting with each other a political battle that has nothing to do with technology may erupt; in any case, performance (and perhaps security) suffers. Provided it has the processing speed not to be a bottleneck itself, a full-feature Layer 3 switch that is capable of all of these functions will make a network simpler and more efficient.

In any event, Managed Layer 3 switches have many more features and are capable of much more than just simple configuration and switching and routing packets. However, if so configured (and this is no simple task) such a device will no longer be a Layer 3 switch but rather an Application Proxy Server with distribution.

Other Networking Devices/Protocols

Other devices are used for internetworking, but we will discuss only gateways here because of their interest to industrial users. The gateway types available include:

- translating bridges/protocol converters
- encapsulating bridges/tunneling gateways
- NOS gateways
- WAN gateways
- application gateways
- site-to-site or computer-to-site VPN gateways

Let's discuss each of these in turn.

Encapsulating Bridges/Tunneling Gateways

We have already discussed the translating bridge (protocol converter)-type gateways. However, the encapsulating (tunneling) gateway type is of some interest to us here. To encapsulate means to surround the originating protocol with the transport protocol. As far as the devices are concerned, everything is just a 1 or a zero. Physical devices put 1s and zeros on the media; they do not care what they represent. The Data Link layer is concerned with the frame organization, but even in the Ethernet frame it does not care how the 1s and zeros are arranged between the type/length and CRCC. Layer 3 looks at its area, yet considers anything beyond its scope to be just 1s and zeros. Even TCP is only concerned with the packet length and sequence number; anything beyond it is just 1s and zeros. So it really doesn't matter what combination of 1s and zeros is placed after Layer 4 (when using TCP/IP). It could be another protocol's entire Layer 2 frame. If the frame is longer than one packet, then it is broken up into fragments and delivered. This is known as encapsulation or tunneling.

Because encapsulation (tunneling), in the end, is how Ethernet and TCP/IP will become the standard in industrial networking. By using a gateway, any protocol can be converted over to Ethernet frames and delivered over the network. This will allow vendors to claim that they use "standard" protocols, even though the resulting signal is usable only on that vendor's equipment.

NOS Gateways

NOS gateways are also called architectural gateways. Most NOS come with several protocols. NT 4.0 had five and can run all simultaneously. This makes it a NOS gateway. Windows 2000/2003 has several protocols, but TCP/IP is the default and the only possible one if Active Directory is used. Novell was a gateway between native IP (version 5.1 and up) and IPX. Unix/Linux has TCP/IP and whatever you load as an additional protocol. All of the standard NOS can run multiple communication protocols as communications stacks.

WAN Gateways

Given the widespread adoption of routing as the preferable method for networking multiple networks, WAN gateways have almost been superseded by the router. Most routers could very well be called gateways. They can mix and match to the WAN, for example:

- Ethernet to frame relay
- Ethernet to SMDS
- Ethernet to ATM

Depending on how you provision them routers generally can connect to fiber optic or copper.

Application Gateways

File format conversions, according to the OSI model, take place in Layer 7. Layer 7 hosts utilities like mail and directory services. Among available application gateways are the following:

- mail gateways
- file format gateways (NFS/FAT/NTFS)
- multi-protocol networking
- directory services

VPN Gateways

VPNs provide a secure channel through a public (usually the Internet) media. Gateways to a network usually come in two varieties, site-to-site and computer-to-site. A computer-to-computer VPN isn't usually considered a gateway type service.

Summary: Internetworking

This chapter has been a brief tutorial on some of the facts you need to understand before starting out on the internetworking path. For an industrial user this is a formidable path, particularly when you have a closed (proprietary) system that doesn't integrate well with the rest of the facility.

Though you can research particular aspects of internetworking on the Internet, most of the information available is either too general or is vendor specific. Multiple reference texts are available, but because of the speed with which technology changes, some topics aren't well documented. Continual study of the literature, upgrade training by vendors, and a good understanding of your requirements and how they are changing are the best ways for industrial users to understand internetworking.

Bibliography

Note that Internet links may change.

Cisco Systems Inc. Cisco CCNP Preparation Library. San Jose, CA: Cisco Systems, 2000.

Doyle, Jeff, and Jennifer Carroll. Routing TCP/IP. Vol. 2. CCIE Professional Development series. Upper Saddle River, NJ: Prentice Hall, 2001.

Microsoft Corp. "Get Connected with Windows XP Networking." http://www.microsoft.com/windowsxp/using/networking/

—-Windows Internet Guide. Redmond, WA: Microsoft Press, 1996.

Naugle, Matthew G. Network Protocols. Signature ed. New York: McGraw-Hill, 1998.

9
Security

Because bad things happen to good computer systems, we've included a chapter on the subject of security and industrial networking. As the hardware and software used in the industrial arena have become far less proprietary and much more like commercial software, security problems have begun to multiply. It is not so much that commercial systems are more vulnerable but that they are much more widely known (which is the reason for their adoption in the first place). Because of the ever present threat of viruses, hackers, back-doors, Trojans, and other malware including spam and those wonderful phishing trips, not to mention testing all of the software patches for operability among various applications, you might think that security was the full-time job of the industrial network technician—and you just might be right.

Defining the Types of Security

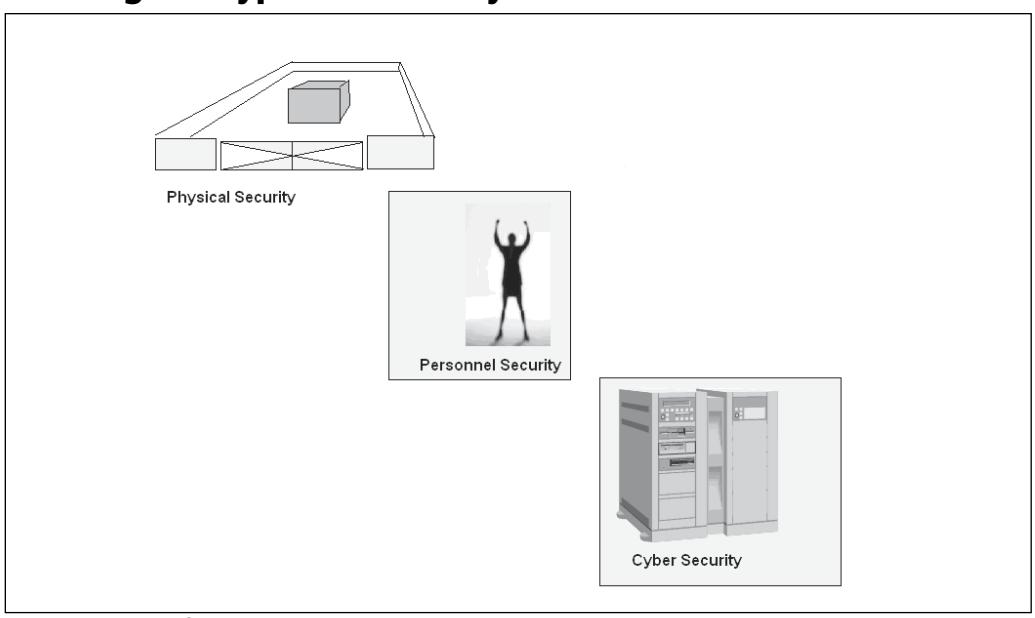


Figure 9-1. Types of Security

Physical security consists of physically ensuring the security of information by using actual guards and gates, vaults, or any form of physical obstruction between the system and a potential intruder. If an intruder has physical access to your system, particularly the servers, the intruder has your information.

Personnel security means ensuring that personnel are not security risks by conducting background checks on employees, having enforceable and enforced written policies for network use and staff conduct, and monitoring personnel for suspicious behavior (like accessing the system when not on duty, etc.).

Cyber security has many aspects, including group policies (a term for policies on a domain controller, etc.), network use policies, firewalls, password policies, and so on. Anything having to do with the computer system itself (access to files, programs, and applications) must be planned ahead and consistent.

This division of security into three areas originated in the security industry. Typically, the general public thinks of security as the first type, physical security. Yet all three aspects of security work together—none is independent. If you only protect one type (cyber-physical-personnel), the potential invader may try another mode to determine if it is unprotected.

The model for modern information security is to preserve the following:

Confidentiality—ensure that information is accessible only to those authorized to have access (the need to know).

Integrity—safeguard the accuracy and completeness of information and processing methods including noncorruption due to malware.

Availability—ensure that authorized users have access to information and associated assets when it is needed.

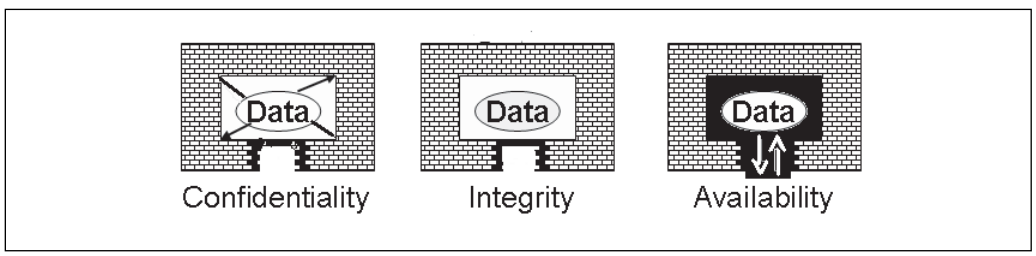


Figure 9-2. Confidentiality, Integrity, Availability

Though industrial and commercial users share these three concerns, physical safety issues as well as the way security is implemented and utilized can be major differentiators between conventional security and industrial security. This is because industrial cyber security must protect lives and facilities as well as data.

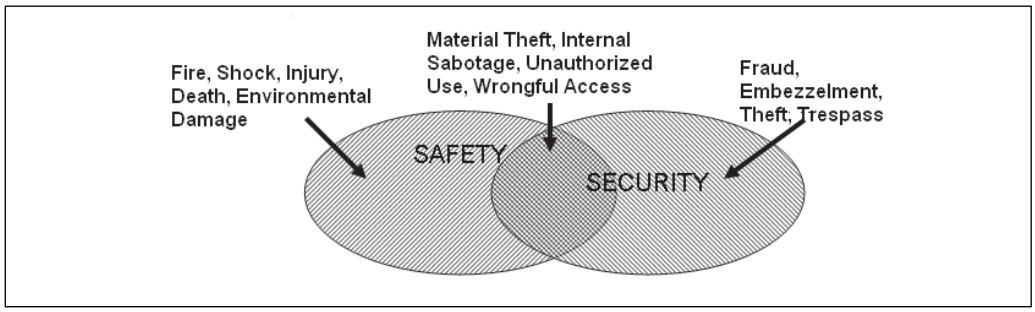


Figure 9-3. Industrial Safety and Security Overlap

Any modern industrial control system may face some security risk, and though that risk may at times be difficult to estimate it must still be accounted for. Though risk cannot be ignored, there is no way to afford perfect security either. Good business practice dictates that a balance be struck between the cost of measures to mitigate a risk event and the potential cost of that event occurring. To strike this balance correctly an industrial facility's risk managers must understand the factors for determining the cyber security risk to the facility.

In a rapidly changing technological climate, a facility must have the capacity to continuously monitor the cyber security risk factors to determine if they also are changing as well as the limits in which they may change. To be effective from both a technical and a cost perspective the actions taken to minimize damage to a facility's assets must adapt to changes in threats, vulnerabilities, target attractiveness, and/or consequences. Let's define some of these terms.

Definitions

Asset: Anything—person, environment, facility, material, information, business reputation, or activity—that has unique value to an owner or adversary.

Consequence: The amount of loss or damage that can be expected from a successful attack against an asset. Loss may be monetary but may also include political, morale, operational effectiveness, or other business impacts.

Countermeasure: An action taken or a physical capability provided whose main purpose is to reduce or eliminate one or more vulnerabilities.

Exploit: Illegal and/or unethical attacks against a computer or system that take advantage of some known or hitherto unknown vulnerability.

Risk: Potential for damage or loss of an asset.

Target attractiveness: An estimate of the value of a target to an adversary.

Threat: Any indication, circumstance, or event that has the potential to cause the loss of or damage to an asset.

Vulnerability: Any weakness that can be exploited to gain access to an asset.

Again, risk reduction must be balanced against the cost of the actions taken to reduce the risk. The goal of a *risk analysis* is to quantify risk. The reduction in risk for a given counter action should include a cost analysis of the countermeasures:

Risk = Threat × Vulnerability × Consequences

As figure 9-4 illustrates, the optimal level of security will be the point where you get the most bang for your buck. Here the cost of security means what it will cost after the countermeasures (if any) have been employed. Note that there is a point of diminishing returns at which the cost of security rises exponentially, yet the level of security (risk reduction) does not.

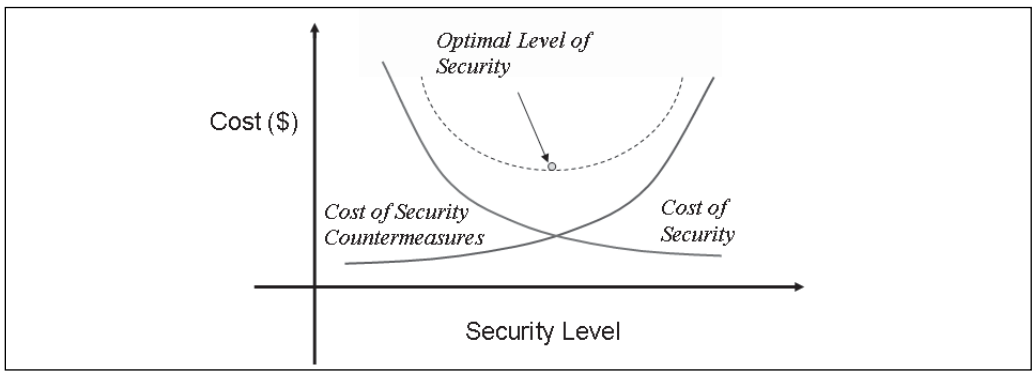


Figure 9-4. Optimal Level of Security

Threats

Recall our risk formula:

Risk = ***Threat*** × Vulnerability × Consequences

What are some of the sources of security threats? They come in two varieties—external and internal. *External* threats include:

- script kiddies (those that modify or copy an original exploit by any of the below)
- recreational hackers
- virus writers and malware in general
- activists
- terrorists
- agencies of foreign states
- disgruntled former employees or contractors

Internal threats can be broken down into unintentional and intentional threats:

Unintentional

- inappropriate access to systems or data
- inappropriate manipulation of systems
- incorrect configurations
- conflicting software
- infected software or hardware

Intentional

- disgruntled employees (presently employed)
- disgruntled contractor (presently contracted)

For external attacks, risk managers might well ask: "we have an industrial control system, and it is not connected to the Internet so how do any of those external threats affect me?" But the system doesn't have to be connected to the Internet to be threatened. It can be connected to any external connection: "bulletin boards," a support group modem connection, even an engineer using PCAnywhere.™ An internal survey of companies using industrial control systems would probably find that most managers believe their critical control systems are not connected to the business network, let alone the Internet. However, an investigation would most likely show that every system is connected in some way to the enterprise (or management) network and that the business network is connected to the Internet using only the security needed to support the business processes. The security employed is usually not comprehensive enough to protect critical systems.

Also, IEEE 802.11 (wireless Ethernet) has become more widely used in industrial settings. It appears that a significant number of systems are deployed without any security features (WEP or WPA) enabled. Though this may just reflect the "just get it to work, we'll secure it later" attitude, somehow "later" just never gets here.

As we stated in chapter 3, even with security enabled, IEEE 802.11 systems contain several well-known security flaws:

- Encryption scheme (WEP) has been and can be cracked
- Authentication scheme doesn't protect the system as a whole
- The MAC layer is susceptible to simple (but effective) denial-of-service attacks

The interim solution is to use the WPA or WPA2 scheme, both of which offer a much higher level of security than WEP. WPA was created by the industry trade group Wi-Fi Alliance. It was designed to be used with an 802.1X authentication server, which distributes different keys to each user. WPA can, however, be used in the PSK (preshared key) mode, in which each user has the same preshared key. WPA is based on the preliminary specifications of IEEE 802.11i.

Data is encrypted using a 128-bit key with a 48-bit initialization stream. One major security advantage of WPA over WEP is the Temporal Key Integrity Protocol (TKIP), which dynamically changes keys during system use. When combined with the much larger initialization stream (over WEP) the inherent key weakness of WEP is removed. WPA2 implements 802.11i. In particular, it introduces a new AES (Advanced Encryption Standard)-based algorithm that is considered fully secure. Official support for WPA2 in Microsoft Windows XP was formally announced in May 2005.

Whether compromising an 802.11 network is considered an internal or an external threat is sometimes open to debate. We will treat it as an external threat in this book, because the network's radio transmissions extend over the facility's boundaries, making them available to personnel outside the facility. These external personnel may not have the company's best interests at heart.

There are several other subtle ways in which outsiders can become insiders. One is the use of laptops. Laptops tend to move between secure and insecure areas, and laptop software tends to be less controlled than desktop systems. A laptop can become infected while it's in someone's home and then bring a "Trojan" into the secure work environment. Once inside, the attacker/virus can take advantage of operating behind the company's firewall, transmitting information outside of the company as an authorized user. Key logging programs will surrender user passwords, and eventually the intruder (unless caught) will have compromised the entire system's safety, performance, and confidentiality.

The "flash," "pen," "thumb," or USB drives, which though small in size are large in capacity (2-8 Gigabytes) are as susceptible to viruses and Trojans as any floppy or hard drive, and they are portable (which is the reason they are used). A major system can be compromised by the use of these drives.

Source of System Flaws

The flaws that attacks can exploit reside in many systems and hence are well known. The reasons why they are well known include:

- The use of common information technology (IT) systems in SCADA such as Windows-based PCs as workstations and SQLServer for database and archival services.
- The use of common networking technologies in DCS, SCADA, and most industrial systems such as TCP/IP.
- The use of embedded systems not originally designed for security
- The use of off-shore development for critical software

Security flaws in major operating systems account for most instances in which a device is taken over. Most control systems rely heavily on Microsoft Windows or Linux, and flaws in these operating systems are well understood by hackers.

Security patches are a necessary evil. Given the complexity of modern operating systems and their interoperation with applications, patches may fix one problem but cause an application to fail (particularly if it is a little-known or custom application). Patch management is complicated for the industrial user by control and safety issues. As an example of this complexity, try installing Microsoft's Internet Explorer 7, a free download. You will find that some settings are irreversible (cannot be uninstalled) except through registry editing, and that it works very well for most environments. Yet I found several script errors, including one that wouldn't allow me to enter a zip code when ordering an item. Now this is not the end of the world; yet suppose the flaw involved a safety or control issue. I am sure that IE7 was thoroughly vetted and went through numerous betas (a more rigorous process than most security patches go through). Yet this is the very reason that companies that use standard OSs and/or applications have to ensure that in every conceivable instance the patch does not break their application. This is why you pay "maintenance" fees to DCS or SCADA vendors who do their own testing and issue their own patches.

If you don't install the patches, viruses can allow Windows- or Linux-based HMI, programming stations, and the like to be taken over. Trojans allow viruses to penetrate through firewalls by masquerading as a legitimate user.

If standard operating systems have security issues it should also be noted that SCADA and control systems were originally designed for performance, not security. Their IP stacks are usually minimal and lack the error checking and recovery functions found in a fully implemented stack. As a result, certain PLCs fail while being port-scanned (which is a typical operation by a hacker or vulnerability checker). Such failure would indicate a serious TCP/IP implementation issue. Many PLCs have legacy commands still deployed on them that are very dangerous from a security standpoint, and nearly all PLC/DCSs have authentication schemes that require only a password for access.

Some Security Needs in Industrial Systems

Few engineers worry about the risk of an outside entity intercepting and acquiring process data. Many would ask: "What possible use could anyone make of some process data?" Unfortunately, hacking may be industrial espionage. Given the fluctuation of gasoline prices in this part of the century, it is not hard to imagine an operator of a fuel tank farm wanting to know what margin he can make when he has to quote an urgent large order request. Wouldn't it be useful to know the status of the other tank farm(s) in the region and what they are quoting? How useful would it be to just get a daily printout of competitors' margins and wholesale prices? Clearly, in the global economy, it is important to protect private process data to safeguard commercial interests, sales, trade secrets, and anything that would aid a competitor.

One of the most common (because easiest to do) exploits is the "denial of service" (DoS). This is an attack that is designed to render a computer or network incapable of providing normal services. It is deadly to a process control network (and quite possibly deadly to all those around said process also). A denial-of-service attack can be performed in several ways. *Bandwidth attacks* flood the network with such a high volume of network traffic that all *available network resources* are consumed, and legitimate user requests cannot get through. *Connectivity attacks* flood a computer with such a high volume of connection requests that all available *operating system resources* are consumed and the computer can no longer process legitimate user requests.

The risk of being unable to view or control a process or system (due to a DoS attack or any other intrusive, invasive, or performance-altering attack) forces facilities to place great reliance on emergency and safety systems. Traditionally, these systems have been separated and made independent of the main control system, and are generally considered "bullet proof." Regardless of the environment and situation they will work as they are supposed to work. However, in following the current trend in the design of the main control systems, these systems are becoming based more on standard IT technologies (such as TCP/IP). Because of the flexibility of software aided and abetted by communications (not to mention

the cost savings) these fallback systems are increasingly being connected to or even combined with the main control systems. This, of course, increases the potential risk of a common mode failure of both the main control system and the safety systems. For these reasons, the risks of cyber attack need to be considered not just when designing control systems but safety systems as well.

Social Engineering

Another form of external attack is "social engineering," in which an attacker solicits information from the victim who gives it to the attacker without suspecting its actual purpose. As an example, suppose a firm e-mails you with an incentive to answer an industry-specific questionnaire that asks you to divulge operating systems, types of equipment, firewalls deployed, and so on. Except for your facility, no one needs to know this information. Some legitimate firms would like to know this information to tailor their bids, but by and large most such requests should be ignored.

Another example: an employee out in the field has lost his contact information and calls another employee to get the remote access phone number. Does the employee receiving the call actually know who is calling him? What could the caller glean using this remote access phone number? Why are persons giving out such private information without authentication?

Another distressful example is "phishing." Here, an e-mail arrives with the proper letterhead, web addresses, and so on, and says something to the effect that your account needs to be refreshed or substantiated and you need to respond with your user name and password. For phishing involving financial institutions, credit cards or Internet banking information (user name, PIN, credit card number, etc.) are elicited. Respondents to these authentic-looking but entirely false e-mails will find themselves victims of identity theft or their accounts will be plundered. In industrial systems, the e-mail would apparently come from corporate headquarters' IT, asking for password and user name.

May it be stated here (and repeated regularly) that no one needs to know your password. IT cannot see your password; they can only give you a new one. *No one EVER needs your password*.

Web-Based Research

One method attackers use to glean information from public sources is generally used before they determine what type of intrusion to employ. This involves using numerous sources of data to find out who owns your system, who has that IP, etc. (Just type your name into an Internet search engine, tell it to GO, and marvel at the results.) Here are a few examples of these data sources:

Internet Network Information Center (InterNIC): "Whois Search" database. The following is a typical "whois" reply from (www.internic.net/whois.html)

Company Name: Registrar used to register domain name
Registrar website: (www.yournetwork.com)
Company Name:
Telephone numbers for locating modems behind firewall and those with unsecured remote access
Contact and email for social engineering
DNS Names servers: DNS Information

Still using "whois"
American Registry for Internet Numbers (ARIN):
Company Name
All IP addresses assigned to that organization.

Query using "nslookup" a command line utility, entering this will return
DNS Server Records:
Domain Name:
Specific IP address
Host System Type of Domain Name

Tools for Investigation by an attacker

Feature	Purpose
Ping (ICMP Echo Request) (built into most TCP/IP suites)	To check if an IP is alive and what its response time is. Ping can use the DNS by using Internet name instead of IP number
Whois	Whois database lookups will provide information on the owner of IP, when IP was established, contact number.
DNS (Domain Naming Service – largest distributed database in the world)	Contains all DNS information about a given domain and can be accessed by nslookup and other hacker tools.
Traceroute (tracert – usually provided with the TCP/IP suite)	Lists router hops (with IP/DNS name) between source and target.
SMTP VRFY	Determines whether email address is valid.
Research and Attack Portals	Similar to Sam Spade software (which operates on an individual machine) but allows users to use the portal via a browser. Traffic appears to come from the Web Server instead of the client machine.
Hacker Tools	Put "hacker tools" in your search engine. It would be advisable to do this on a stand-alone machine that is not essential to the plant and has no useful information on it. Many times when you download from these sites, hacker insertions, viruses, Trojans, and malware galore come with your download.

Table 9-1. Research Sources

Password Cracking

A system's main protection (from both external and internal attacks) is the password. If you think passwords provide adequate security you may also believe in the tooth fairy. The problem simply stated is this: people are human. An effective password would be a string of thirty-two random alphanumeric—upper and lower case—characters with punctuations. But who could remember that? It would need to be written down. And where will it be kept? On a sticky note on the monitor or, if the user is exceptionally security conscious, on the bottom of the keyboard?

The only way a password like this will be utilized is through system dictates. If the user is allowed to generate his or her password it will be short and memorable. Password-cracking systems know this. Quite often passwords and user IDs are:

- identical, usually the user's first or last name
- system defaults ("sa" for SQL Server, blank for XP Home users)
- easy to guess, such as a company name, variations of user names or TV characters, particularly Star Trek or other techie shows
- guest accounts or active terminated accounts
- found on sticky notes attached to the monitors

Automated hacker tools, such as Crack, L0phtcrack, and John the Ripper, tend to brute-force entire dictionaries but can quickly determine short and common passwords.

Internal Threats

The greatest internal threat is a disgruntled employee. Most intentional unauthorized uses are primitive and include hacking behind the firewall, sabotaging files and applications, inserting time bombs (a malware program that initiates long after the employee has left), and password theft. And most usually come from an employee who already has access to the system. This is a new problem only in the problems technological implementations.

Other internal threats are not intentional and generally are caused by a faulty system design (i.e., not idiot proof), lack of communication, improper training, curiosity, and pure accident. These are minor (unless an incident of this type brings your plant down), and intentional threats are far more prevalent (particularly when performing risk analysis). Remember:

Risk = Threat × ***Vulnerability*** × ***Consequences***

Risk Analysis

Assessing the value of an industrial cyber attack is not simply a matter of assigning a financial value to an incident. Although obvious direct financial consequences may be easily quantifiable (e.g., loss of production or damage to plant), other consequences may be less obvious. For most companies the consequences of an attack for damaging the company's reputation may be far more significant than just the cost of a production outage. The impacts of these incidents on health, safety, or environment could be serious to a

company's brand image. Even consequences as minor as a limited regulatory contravention may impact a company's reputation or, possibly, its operating license.

Risk Analysis Steps

There are five steps to risk analysis:

1. List the assets to protect
2. Identify threats
3. Establish vulnerabilities
4. Rate the risks
5. Select countermeasures to mitigate risk

Step 1 is self-explanatory, and since we have already discussed step 2 we will now consider step 3, Establish vulnerabilities, here. This step is done by determining what the control system's vulnerabilities are and what they could be. The following lists are typical, not inclusive, of system vulnerabilities. Most systems undoubtedly have other vulnerabilities.

1. Establish the control system's vulnerabilities:
    - Internet/intranet connections
    - remote access
    - laptops and USB drives
    - dial-up modems
    - wireless communications
    - theft of handheld devices or laptops
    - OEM, vendors, and third-party access
2. Examine the system environment for control system vulnerabilities:
    - poor password management
    - gullibility of people to social engineering
    - system default configurations
    - lack of computer security policies
    - lack of communications with Human Resources, Contract Management, and other affected departments
    - lack of computer security training for personnel
    - inappropriate trust relationships between domains

In order to rate these risks, you must determine their possible consequences. This list, like the first two, is not inclusive but typical. Your consequences will depend on the system, processes, process materials, environment, and so on.

The possible compromised system consequences may include:

- endangerment of public or employee health and safety
- customer safety risks
- economic or continuity of production loss
- violation of regulatory requirements
- damage to company reputation
- lowered corporate stock price

The lists of vulnerabilities and consequences provided here are based on the ISA course IC32, "Cyber Security for Automation, Control and SCADA Systems." Anyone involved in security for automation systems should attend that or an equivalent course. Table 9-2 is developed (from examples in that course) and is a typical small segment of a complete table.

There is just one threat listed, and apparently this is threat 1.001 of many.

There are two listed vulnerabilities that could bring about this threat.

There are four consequences listed, two for each vulnerability. And there are four listed safeguards, one for each consequence and states that the possible consequences cannot be obtained off-site due to the limited energy or toxicology of the process.

There are, of course, recommendations to enhance the safeguards and reduce the vulnerabilities and with those actions the responsible department….

Threats	Vulnerabilities	Consequences	Safeguards	Recommendations	Action
(1.001) Manipulate PV/SP by remote access, causes process upset	(1.) VPN from vendor	(1.1) Possible on-site fatalities	(1.1) SIS independent of main system	(1.1) Restrict physical access to VPN equipment	HR
	(2.) Remote access by modem	(1.2) Possible off-site fatalities	(1.2) Process upset does not have potential energy or toxicity to cause off-site damage, directly or indirectly	(2.1) Eliminate modem access	IT
		(2.1) Possible on-site fatalities	*(2.1) same as (1.1)*		IT
		(2.2) Possible off-site fatalities	*(2.2) same as (1.2)*		

Table 9-2. Example of Risk Analysis

Countermeasures

Some of the actions that may be taken to help protect an automation system from an internal or external attack include the following:

- Train users about social engineering, how to react, and how to report
- Locate modems and wireless systems before attackers do—ensure security is enabled, that modems use callback; or better, remove all modems behind the firewall
- Define and implement a password policy, enforce it, and train users in it
- Develop Help Desk procedures to authenticate users who are requesting password resets
- Use router-based firewalls to deny most incoming and some outgoing connections through a defined policy
- Using a port-scanning tool scan systems to determine open ports. Close all unnecessary ports (TCP/UDP), uninstall unused software
- Configure router services to allow ICMP Echo Requests (pings) only from the ISP's management systems
- Use an Intrusion Detection System. Run a vulnerability scan against your own network periodically
- Do a scheduled update of user lists, deleting personnel who have been terminated (at the time of notification if possible), are retiring (effective date), or undergoing changes in their organizational responsibilities that give them more or fewer permissions.
- Restrict DNS information leakage:
  —There should be no DNS HINFO and text files concerning Internet-accessible machines
  —Restrict zone transfers (forward or reverse) to appropriate servers
  —Employ a split DNS to exclude internal-only systems information

Firewalls

One of the most effective protections available is the firewall. All firewalls are not equal. There are three major types: packet inspection, stateful inspection, and proxy. Additionally, some firewalls are software based (run as an application or service) and some are hardware (appliance) based.

Packet inspection firewalls basically work at the Network layer. They:

- examine only the headers of each packet of information
- accept or reject each message based on the packet's sender address, receiver address, or TCP port
- perform firewall checks on each incoming or outgoing packet for its source address, destination address, and function (based on the IP port assignment)
- accept or reject packets based on a comparison of packet Layer 3 information to several predefined rules called Access Control Lists (ACLs)

Packet inspection firewalls are typically Layer 3 only and are often router based.

Pros: Very fast
Cons: Examines packet headers only and not the overall session or packet content

Stateful inspection firewalls surpass the packet filter firewall by:

- tracking the relationships between packets in a session
- inspecting the contents of the packet

They can be either router based (e.g., Cisco PIX) or server based (e.g., MS ISA).

Pros: Relatively fast
Flexible
Improved security
Cons: More expensive

Proxy firewalls basically work at the Application level. They

- handle packets for each Internet service by interpreting the command at the top layer
- act as an intermediary that accepts connections and requests from a client and then issues them
- basically interprets every incoming packet up to the Application layer, checks it, and then reissues it to the target device

Proxy firewalls are usually server based:

Pros: Very strong security model
Cons: Slow and process intensive, only handles well-known services

Both proxy and packet inspection firewalls allow private IPs to be used on the user side of the proxy. Accepts incoming and transmits outgoing e-mail, the World Wide Web, chat, and newsgroups, stripping off the information that identifies the source and passing it on to the Internet.

By introducing a simple firewall between the enterprise and process control networks, you can achieve a significant security improvement. Most firewalls on the market today offer stateful inspection for all TCP packets and application proxy services for common Internet application layer protocols such as FTP, HTTP, and SMTP. Correctly configured, firewalls significantly reduce the chances of a successful external attack on the process network.

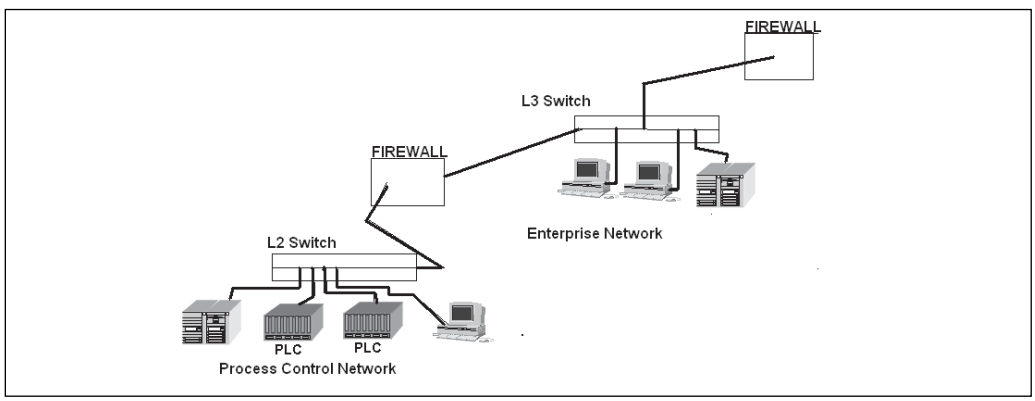


Figure 9-5. Firewall between Control and Enterprise Networks

Commonly shared devices such as a data historian and mail server can be placed in a "demilitarized zone" (DMZ) between the enterprise and process control networks. Each DMZ holds a separate "common" component, such as the data historian, the wireless access point, or remote and third-party access systems. In effect, the use of a DMZ-capable firewall allows you to create an intermediate network, often referred to as a Process Information Network (PIN).

For you to create a DMZ the firewall must offer three or more interfaces rather than the typical public and private interfaces. One of the interfaces is connected to the enterprise; the second is connected to the process network, and the remaining interfaces are connected to the shared common asset.

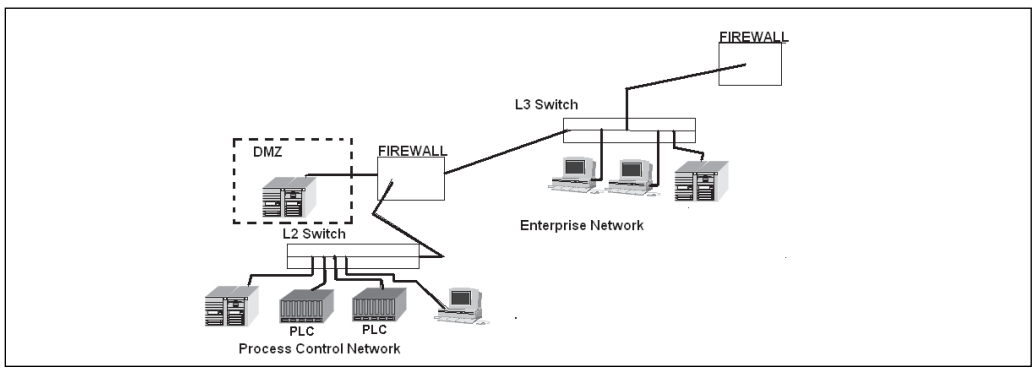


Figure 9-6. Use of a DMZ Firewall

The assets in the DMZ may be equally accessed by either side of the firewall.

Network Address Translation (NAT)

One technique used by proxy servers is Network Address Translation (NAT). Originally designed as a way to conserve IP addresses NAT is now used to hide networks away from prying eyes. There are several "private" IPs. That is, if these IPs would appear on the Internet, the first router encountered would not forward the packet. 10.xxx.xxx.xxx, 172.xxx.xxx.xxx, and 192.168.xxx.xxx are all private addresses. The NAT device merely maps one of the private addresses to a port and transmits packets through the single IP owned by the facility. In its transmission it notifies the receiving station of the reply TCP (or UDP) port used.

Perspective on Security

This chapter has discussed security as a risk concept and has reviewed typical threats as well as how to analyze the threat, its consequences, and countermeasures. This brief explanation should do one thing—ensure that you are aware that security is a complex, essential, and necessary subject, one for which those responsible for security must get further training. As technology changes, as the countermeasures become effective, the threat will evolve into a different form. It is very much like the influenza virus. Vaccines are prepared on the most likely strain, and people are vaccinated. Yet the flu virus seems to be able to morph into dif-

ferent forms against which the vaccine is not as effective as it could be or not effective at all. Cyber malware is the same. It is the price we pay for connectivity to the larger world.

The Internet

We have included the Internet in a book about industrial communications because it has become an indispensable part of network technology. Simply, you will be using Internet technology on an intranet even if you do not use the Internet. The Internet is a wide area technique that is extremely cost effective. The use of the higher bandwidth accesses (DSL or cable modems) allows you to effectively yet inexpensively interface to corporate LANs. Using the Internet creates other problems, the most paramount being security. No one wants to think that a thirteen-year-old could manipulate a setpoint in your plant. But that could happen if you used the network as an unsecured user. However, with the advent of effective and secure cryptography a Virtual Private Network (VPN) will enable corporations to greatly reduce their recurrent communications cost and also reduce the threats described in this chapter.

Encryption

In this section we'll discuss encryption's ease of use and some related safeguards. Coding and ciphers are two ways of making information secret. Codings, however, are just substitute words or phrases that are generally understood by both parties. Ciphers, on the other hand, are generally *composed* using the smallest unique part of the information. For a written letter a cipher would be performed on the individual characters. For electronic transmission the cipher would occur on each bit. A cipher may be of the transposition or substitution type. In the transposition type the individual bits are retained (only moved around according to a key held by both sides). The substitution types substitute a cipher bit for the data bit and then replace it with the original bit upon deciphering. Developed first by Vernam in 1917, the basic method of enciphering and deciphering is illustrated in figure 9-7. A block cipher encrypts on a block of data, this will necessarily mean storing data for a block time meaning it is not "real time" but store and forward. Streaming ciphers on the other hand just run the data bits through an Exclusive-Or (XOR) with one input the data bit, the other the key bit and the output the cipher bit, the reverse is performed upon decryption (see figure 9-7) This requires more demanding circuitry in terms of speed and processing. This does not mean that history and preview cannot be performed on the data stream but that it is done in a register of sufficient length and running at data speed.

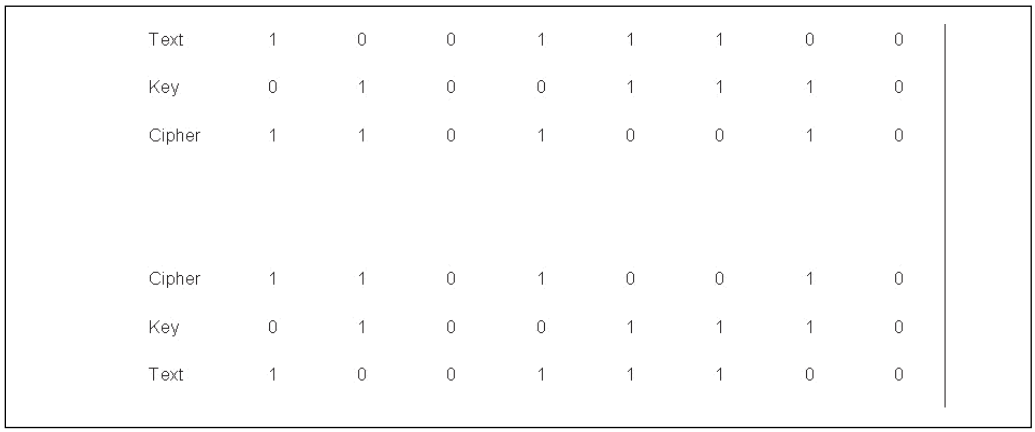


Figure 9-7. Encryption Example

Encryption takes place when the plain-text (message to encipher) is Exclusive-ORed with the key. At the receive end, decryption takes place when the enciphered message is Exclusive-ORed with the key and the plain-text reappearing. It is that simple. What is not simple is generating the key. To be a perfect key it must be nonrepeating. If you lack both the time and equipment to prove that a number is nonrepeating, the next best alternatives are to choose the derivatives of the prime numbers (a prime number can only be factored by itself and 1). Therefore, if you find a large enough prime number, the results of its repeated division will not repeat themselves. Electronically this is duplicated as a pseudo-random number string. It is pseudo because it has to be reproduced in the receiver as it is in the transmitter, and the chance of two very long nonrepeating numbers being the same by chance is somewhat small (much like winning the lottery).

In a simple point-to-point circuit or link the key is limited to both ends; no one else should have it. It would probably be distributed by physical means. Keys must be changed after a specified period of time or they are vulnerable to being broken, which is known in the business as being "compromised." This is because the key really isn't an endless nonrepeating number and because the patterns for most primes are known. Irrespective of the complexity with which the basic pattern was modified, if the key is used long enough it is vulnerable to attack (particularly by computer). This is especially true if a weak enciphering system was used to start with.

Breaking a cryptographic cipher has many facets, and the use of ever more powerful computers and networks of computers has greatly aided the task. Letter frequencies, business environment, network functions, traffic flow patterns, and cultural uniqueness all contribute to the analysis of an encrypted message. The single most effective item in compromising a cipher is a plain-text (not encrypted) source data whose time and location of transmission can be identified. With this information and the data of only a few packets all messages used during that key period (or in the case of a public key, many periods) can be disclosed.

Though most business data requires time stamps, if your network data has been encrypted (or will be) and the data is to be stored before transmission, it should be encrypted to deny attackers the opportunity to compare the plain text stored data with the encrypted transmitted data.

In cryptography, the key is key. How can the keys on a local area network be distributed securely, particularly if there are a number of remote sites? And the question is even more complex if each site has multiple levels of access. Actual physical distribution is costly and inefficient. Keys that must be distributed in this manner are known as private or secret keys. They cannot be distributed electronically (on the same network as the previous key) as there is a chance the previous key may be compromised. Keys broadcast on a network are known as public keys. Typically, a combination of private and public keys is required for electronic distribution. In the private key system, the same key is used to encrypt and decrypt the data. In a public key system pairs of keys are used. These keys, a public and a private key (at each node), provide a key algorithm. One may be derived from the other; however, this is not reversible. One of the pair on the transmit end encrypts (transmission end), and the other of the pair decrypts (reception end). This is the basis for digital certificates, Secure Sockets Layer (SSL), and other Internet securing methodologies.

Different levels of access require different public (and private) keys. Notwithstanding its ease of distribution, a public key is usually less secure than a private key if only because it is publicly distributed, and everyone has access to that distribution. That being said, one would have to wonder what information most companies (not involved in the defense sector) would have on a network that would prompt such an expensive attack, since the cost of determining the public and private keys would be high. Paranoia aside, a public key system properly administered will efficiently and securely protect most industrial and business internal LANs from external threats. Many companies offer encryption protection for files and transmitted data. This protection usually takes the form of hardware adapters, which are quicker than software, although effective software does exist and is increasingly built into operating systems' environments.

The Digital Encryption Standard (DES) was a standard key that has now been replaced. It was a block cipher, that is, an output block that was enciphered as a function of the input block and the complete key. In the DES, a 64-bit block is encrypted by a 56-bit key. An adapter with a VLSI chip usually performs this enciphering because software (until the advent of faster microprocessors) was just too slow. Most financial transactions by financial institutions and Federal Reserve banks were conducted using this standard. DES is considered a 16-bit standard. One problem for those considering using DES is that it requires a synchronous data stream. The PC is asynchronous. Some method of conversion must therefore be employed, usually an adapter card that has a number of hardware registers and a synchronous modem if it is to communicate over dial-up lines. DES uses private keys. The actual security of the DES was always held in some suspicion by some people because of its

origin—IBM and the U.S. government. This could normally be dismissed as the usual outcry of the small anti-government minority, except that the U.S. government does continually press for the integration of backdoors (a method of defeating the encryption process) into any encryption process. This requirement was ostensibly for maintaining law and order; using the backdoor could only occur by court order. Indeed, at one time the U.S. government was considering requiring the installation of an encryption chip with a specified backdoor in every computer used by government or private business facilities. Some good encryption programs are available, notably PGP (for Pretty Good Privacy), which is administered by RSA (a company dedicated to developing and selling privacy products).

The DES was quickly broken by many networked computers, so it has given way to a new standard encryption method: Triple DES (1,024 key bits). The trick is to make the key long enough and secure enough that the attack takes longer and costs more than the information is worth. The U.S. government actually sponsored a challenge for the development of the standard algorithm for an Advanced Encryption Standard (AES) that drew several notable contestants. Triple DES was one of them, but it did not win, although the government stated that any of the competing algorithms was sufficiently safe for modern data transmission.

Actual encryption for one or two parties is quite simple; it is on the networks that it becomes complex. If you would like to encrypt a single file, do the following: use one of the many file encryption utilities, use the maximum length key, and ensure the key is a random assortment of alphanumerics. After encrypting your file, encrypt it again with a different long, random, alphanumeric key. No one can recover your file through cryptographic analysis if the file isn't too long in relation to your key size. So in this example, a file of about 4 Kbytes and a key length of 256 octets would be secure against all but a supercomputer attack.

The problem is this: Did you write the keys down? If not, you won't be able to recover it either. Now consider a LAN with a large number of nodes. How are all these keys to be distributed, changed, and so on? This is the problem with ensuring that your data will withstand all attack. Less secure systems cost less, are easier to implement, and will probably easily meet the real threat to your data. You can spend a lot for security that is not needed. However, when your information is placed on the Internet, then it is fair game to all the denizens of that world. Strong encryption is a necessity. SSL, PGP, Digital Certificates; they all have one thing in common: long keys.

The Internet technologies (n-tier) are now being employed in the industrial areas. In fact, one particular candidate for a fieldbus actually gives each device (sensor, actuator, whatever) its own web page. The information available to the operator is real time, and all the actual operating specifications are there to see.

At present, the low costs associated with the Internet make it a good bet to replace a large number of lower-speed WAN lines (this fact is not lost on operating companies or govern-

ments). Since you can transmit anywhere in the world for the cost of an ISP and such at each end, you can rest assured the Internet will be used for industrial WAN replacement. Include the fact you can make it secure (with a VPN) and you have the recipe for cost-effective communications.

A VPN is a private network that operates on a public infrastructure. Typically, VPNs provide encryption "tunnels" for data over the Internet. You use the Internet for transport, but your data is indecipherable to all but the intended group.

There are three classifications of VPN deployments—site-to-site VPN, remote access VPN, and host-to-host VPN. Each VPN deployment uses two types of devices: security gateways and hosts. The two endpoints of the VPN are intermediary devices. They pass traffic from a trusted network to another trusted network while relying on the VPN technology to secure the traffic on the untrusted transport network, an arrangement that is commonly called *site-to-site or LAN-to-LAN VPNs* (see figure 9-8).

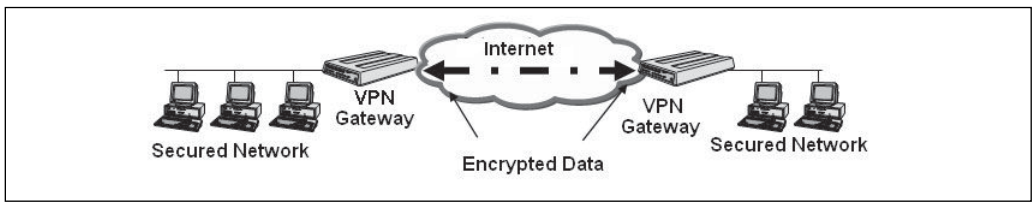


Figure 9-8. Site-to-Site VPN

In *remote access VPN*, figure 9.9, one endpoint is a host computing device, and the other endpoint is an intermediate device that passes traffic from the host to the trusted network behind the security gateway while relying on the VPN technology to secure the traffic on the untrusted network.

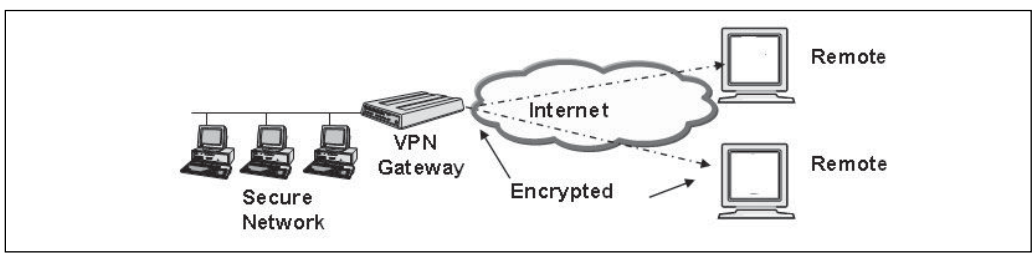


Figure 9.9. VPN to Host

In host-to-host VPN, each endpoint of the VPN tunnel is a host computing device such as a PC.

The host devices use VPN software on the computer to secure the communications on the untrusted network.

There are many technologies for building a VPN over the Internet (or intranet). The dominant ones are:

- Secure Sockets Layer (SSL): SSL is easier to deploy but less standards based. Installing an SSL client requires little or no administration.
- Internet Protocol Security (IPsec): IPsec is harder to deploy, but is based on IETF standards. Achieving interoperability between vendor IPsec implementations can be difficult. IPsec is included in many operating systems.

VPNs were designed to use key management from a PKI (Public Key Infrastructure). However, they can use a preshared private key in lieu of key management. If a preshared private key is used, both ends need the identical phrases to initiate the VPN tunnel. Use as long a phrase as possible, and one that's random and alphanumeric with punctuation. In no case should it be less than sixteen characters.

When given the choice use either 3-DES or AES for both IKE (Internet Key Exchange—IKE transparently negotiates encryption and authentication keys) and VPN communications and use as long a key (1,024 characters or higher) as possible.

Network Security Management

Network management could be the subject of its own book, and many quasi-automated management programs are available. To be accurately and completely performed, network management requires software assistance. In the following pages we'll look at these aspects of network management:

- configuration
- security
- error/fault handling
- accounting
- performance
- redundancy

WAN/LAN management is a must for network management performance. Good network operation depends on good management. Connectivity, security, performance, and available resources are all part of network management. All management schemes will add overhead traffic to a network. Though many management schemes are in general use, most provide only partial implementations of most companies total network management goals which would include all 6 topics listed previously.

Network Security Configuration

Configuration is essential to implement network security connectivity; change, add, and delete users; and change the logical order of station addresses and routing assignments. Additionally, configuration includes adding or deleting access and/or functions to a station, bridge, router, or gateway as well as managing equipment and node features, including software provisioning.

User authorization is another name for multilevel security, which is generally done by passwords. Different users in an organization have differing needs. Not everyone should be an administrator (although that does simplify access, it is hardly a suggested procedure). Users should have only the access they require, no more. Most major NOS provide for not only differing levels of access, but also designated time windows when the user can log on as well as frequency limits controlling how many log-ins the user may have at one time. Caution should be observed in that most security systems use the lowest security assigned to the user regardless of which groups the user is placed in (in fact, the whole group may be denied if a member who is disallowed is assigned to it).

Key management is how you store and distribute your encryption keys and digital certificates. Given that encryption is the most effective way to keep your data private and invulnerable to tampering, it will soon be pervasive even in industrial areas. Your liability would be quite high if your security could be compromised through your data transmissions and the resulting damage caused great monetary damage or loss of life. Encryption is a well-thought-out set of processes and has just now made it into the cost-effective and non-performance-limiting arena. Of course, if you do not effectively manage your keys and their distribution then you may very well compromise your system.

Access management is generally the task of the system administrator. There is and will be both technical and managerial problems as industrial systems are integrated into the commercial and office systems. For a start, who has access control over the industrial network? Wrong settings or poor response time to a request for access changes could be extremely detrimental to system operations. Who will determine your technical requirements? An industrial measurement and control system has a host of requirements and commitments different from any office system. Who allocates bandwidth and connectivity, and who dictates platform?

Network Error/Fault Handling

Because faults will occur you must have the following information to ensure errors are handled correctly:

- node status
- backup and restore procedures
- methodology, reporting, and alarming of station, server, system error handling
- procedures and location for logging and auditing of errors
- system error tolerances

Generally, some form of software program will detail all these for you. In most respects, industrial systems will normally have these types of data available for you as a safety and performance requirement.

Network Accounting
Network accounting is used to:

- apportion traffic cost
- determine license use
- determine software use
- determine user costs

The system administrator performs most of these efforts. Again, the system administrator must understand the industrial system requirements before setting policy.

Network Performance
Network performance is monitored to:

- determine instantaneous network status
- document historical record
- analyze traffic
- correct bottlenecks
- track users' apparent speed (the response speed to a user's keystroke is important for eliminating multiple key strokes, user interventions, user panic, or other debilitating user actions caused when he or she thinks the system is inoperable when it is just slow)

Summary

This chapter has covered a wide gamut of network knowledge, from security to network management software. All of the topics discussed here change on a daily basis as technology moves forward. So each topic in this chapter will require you to give it further study and continually upgrade your knowledge.

Bibliography

CSPP Guidelines for Analyzing and Managing the Security Vulnerabilities of Fixed Chemical Sites David Moore, PE, CSP AIChE, August 2002

ISA. *IC32 Cyber Security for Automation, Control and SCADA Systems*. Research Triangle Park, NC: ISA, 2005.

National Infrastructure Security Coordination Centre. "The Electronic Attack Threat to Supervisory Control and Data Acquisition (SCADA) Control & Automation Systems." London, UK: NISCC, July 12, 2003.

—-"The NISCC Good Practice Guide on Firewall Deployment for SCADA and Process Control Networks." London, UK: NISCC, 8 July 2004.

Prologue to Fourth Edition

In the first two editions of this book, I spent a bit of time prophesizing about the state of industrial networking. Although I find I was right more often than wrong, I also found that my prophecies didn't really make any difference and so stated this in the prologue to the third edition. There is now an international Fieldbus standard that includes the specifications for Foundation Fieldbus H1 and HSE, Profibus PA, DP, Profinet, WorldFIP, ControlNet, Ethernet/IP, and many others. These are not interoperable in any way, but they do share a common format for the standard. The combination of all these specifications enables and promotes international free trade, making this combined specification a political standard, not a technical one. However, the marketplace will decide who gets the lion's share of business. I will predict that whoever makes the best use of object-oriented programming, makes the user interface easy, and makes it easy to purchase, install, and maintain will be the winner.

Technologies to watch will be the continuing transition to Web standards used in lieu of custom interfaces. Wireless technologies will be of great benefit in many industrial areas as various problems related to the technology are resolved. Industrial Ethernet is increasingly the winner in the Layer 1 and 2 areas, and TCP/IP (for Layers 3 and 4) will be used even in the industrial areas, along with all the attendant security and timing problems. Cyber security will be of increasing importance throughout the enterprise.

Additionally, you will find three appendices at the end of the text. These are an informational and sometimes historical review of the fundamentals supporting data communications. Some of the material, it is assumed, a reader of this text might be aware, but if that is not the case, the appendices are there to help.

Questions about this text or any information (including where the author can research it) should be directed to the author at larrymthompson@hotmail.com.

Appendix A
Number Systems Review

This appendix is concerned with number systems, primarily the binary number system. To understand digital systems, an acquaintance with the binary number system is necessary (however, it should not be terribly painful). While becoming familiar with binary, some of the other systems of interest will be brought up, such as "hexadecimal" systems. All of these find use in data communications systems.

The Decimal System

All number systems follow the same rules. You are probably quite familiar with one number system, namely the decimal system. Decimal means that the number system is based on 10 digits. Its base is 10. The only numbers allowed in the decimal system are 0,1,2,3,4,5,6,7,8,9. All the numbers that we can use to describe numerical quantities in decimal are made up of those 10 numbers and no others.

Figure A-1 illustrates the powers of ten. Notice that a number, such as 4,302.63 is really 4000 + 300 + 2 +.6 + .03 or more correctly, 4 × 1000 (10 to the 3rd power) + 3 ×100 (10 to the 2nd power) + 0 ×10 (10 to the 1st power) + 2 ×1 (10 to the 0 power or 1) + 6 ×1 X 1/10 (10 to the –1 power) + 3 ×1 ×1/100 (10 to the –2 power).

1 X 1000	1 X 100	1 X 10	1 X 1	Decimal Point	1 X –1/10	1 X 1/100
4	3	0	2	.	6	3
4302.63						

Figure A-1. Powers of 10 Example

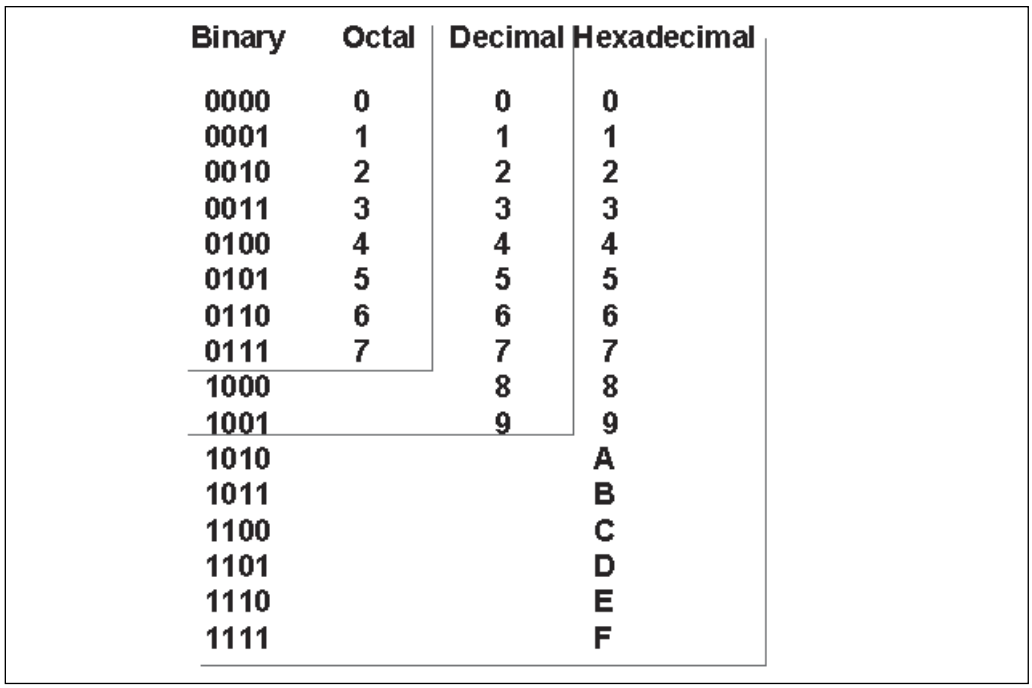
All other number systems will be constructed the same, except rather than 10 as the base, they are based on a different number. For example, the base will be 2 if binary or 16 if hexadecimal.

For this review of number systems, it is important that the emphasis be on the use of the number system as a means of pattern recognition. The number systems chosen here are the ones used presently in data communications. Binary is the number system used by computers; decimal is used to represent binary values so they make sense to humans; and hexadecimal is used to reduce binary streams to recognizable patterns. Computers presently only work with only binary patterns, and humans really understand only decimal.

The arithmetic functions for the binary system will be explained next.

The Binary System

Figure A-2 illustrates the binary values for 0 through 15 decimal.



Binary	Octal	Decimal	Hexadecimal
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000		8	8
1001		9	9
1010			A
1011			B
1100			C
1101			D
1110			E
1111			F

Figure A-2. Binary Numbers 0 - 15

The binary patterns for the decimal values 0 through 9 are called binary coded decimal (BCD). This means that the decimal numbers (0 through 9) are each represented by four binary digits.

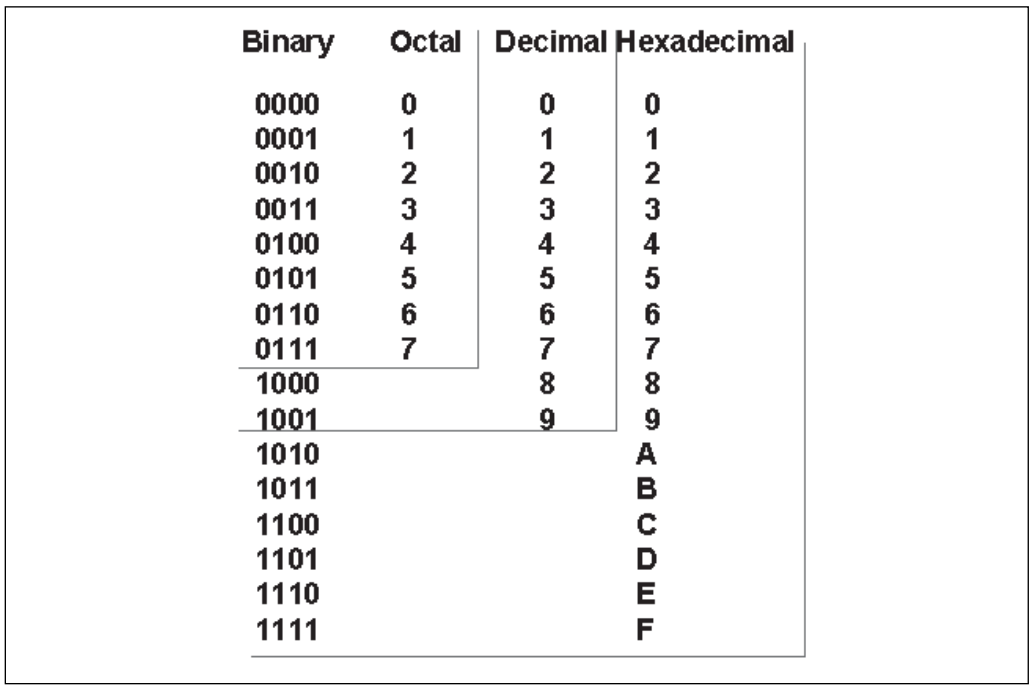
1	3	0	2	DECIMAL
0001	0011	0000	0010	BCD

This was done because humans understand decimal, not binary. BCD coding is used to represent decimal values in a binary format.

Early computers used a 12 bit "word"; therefore, breaking the word up into four 3-bit patterns would allow representation of each of the three bits by its BCD value. Four-bit patterns were not desirable since using a leading "1" (BCD digits 8 and 9) wastes 6 patterns (10-15) as the patterns must be represented by a unique single digit. Since the BCD coding for 0 through 7 only uses three bits, and has 8 unique patterns, it is called "octal." Conversion from binary is performed by separating the binary number into groups of three, starting at the binary point. Assign the BCD value for the 3-bit group and you have performed the conversion.

The Hexadecimal System

Hexadecimal number systems are based on 16. The binary coded decimal arrangements are shown in figure A-3:



Binary	Octal	Decimal	Hexadecimal
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000		8	8
1001		9	9
1010			A
1011			B
1100			C
1101			D
1110			E
1111			F

Figure A-3. (Repeat of Figure A-1s)

Note that this is the same arrangement as used to illustrate BCD, except that patterns that were illegal in BCD (10 - 15) are assigned unique single character representation for hexadecimal. Therefore, the 6 patterns that were wasted can now be used. This lets us represent 16 unique 4-bit binary patterns. Modern computers perform all operations on 4-bit or some multiple of 4-bit patterns; therefore, hexadecimal representation is most often used.

Conversions

Binary Pattern to Hexadecimal

First separate the binary pattern *starting from right digit* into 4-bit groups. Assign the hexadecimal representation to each group.

EXAMPLE A-1

Pattern: 0111010010100101
Separate into groups of four from the right-most digit:

0111	0100	1010	0101
7	4	A	5

Hexadecimal Number to a Binary Pattern

Write out the 4-bit patterns for each number.

EXAMPLE A-2

Hexadecimal numbers are generally written as 0hexnumbersH, leading 0, following H)

0FF13H (FF13 is hex representation)

F	F	1	3
|	|	|	|
1111	1111	0001	0011

1111111100010011

Decimal to Hexadecimal, Hexadecimal to Decimal

(If frequent conversions are required, an inexpensive calculator should be used).

To convert from decimal to hexadecimal:

1) First convert to binary
2) Then convert to hexadecimal

The following patterns should be memorized, both decimal and hexadecimal, because they are among the most common patterns encountered.

0 - 15 (0 - F Hex) listed previously.

Table A-1. Popular Conversion Numbers

Binary pattern	Decimal	Hex
1 0000	16	10
10 0000	32	20
100 0000	64	40
1000 0000	128	80
1111 1111	255	FF
1 0000 0000	256	100
10 0000 0000	512	200
100 0000 0000	1024	400
1000 0000 0000	2048	800
1 0000 0000 0000	4096	1000
10 0000 0000 0000	8192	2000
100 0000 0000 0000	16384	4000
1000 0000 0000 0000	32768	8000
1111 1111 1111 1111	65535	FFFF

To convert a binary number to decimal (and then to hexadecimal), you use the powers chart (much like table A-1).

Power of two															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Decimal Value															
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
Sample binary number															
0	0	0	0	0	1	0	1	0	1	0	1	0	0	1	0

Simply add up the columns that have a 1

1024
256
64
16
2
1362 is the decimal equivalent of 10101010010

To convert a decimal number to binary, perform a procedure much like long division:

Example A-3

Convert 953 to binary.

Problem: Using the powers of two chart, what is the largest power of two which will go into 953?

Solution: It is 512. Using a 12-bit binary number means all of the higher bits until 512 will be 0s.

Power of Two											
11	10	9	8	7	6	5	4	3	2	1	0
Decimal Equivalent											
2048	1024	512	256	128	64	32	16	8	4	2	1
Binary Number											
0	0	1	1	1	1	0	0	0	0	1	1

512 + 256 = 768. This is less than 953 so put a 1 in the 256 column.

512 + 256 + 128 = 896, less than 953, put a 1 in the 128 column.

512 + 256 + 128 + 64 = 950, so keep a 1 in the 64 column.

512 + 256 + 128 + 64 + 32 = 982 so put a zero in the 32 column.

512 + 256 + 128 + 64 + 16 = 966 so put a zero in the 16 column.

512 + 256 + 128 + 64 + 8 = 958 so put a zero in the 8 column

512 + 256 + 128 + 64 + 4 = 954 so put a zero in the 4 column

512 + 256 + 128 + 64 + 2 = 952 so keep a 1 in the 2 column

512 + 256 + 128 + 64 + 2 + 1 = 953 so keep a 1 in the 1 column

The resulting binary pattern is 001111000011, and prepared for HEX it is:

0011	1100	0011
3	C	3

You will find it necessary to use these conversions when in learning situations (trying to understand equipment operation, etc.), diagnostics (problem location with differing equipments of different manufacture), and software/programming.

If you find yourself having to move between decimal and binary (or hexadecimal), it is best to use a calculator; it is far less time consuming, usually a great deal more accurate.

Conversion between hexadecimal and binary only requires knowledge of (or memorization of) 16 patterns, 2, of which are the same for both (0 and 1).

While this is not an extensive investigation into number systems, or even one number system, it should be a sufficient base from which to manipulate between the most commonly used number systems.

Appendix B
Historical Aspects

Introduction

Modern industrial data communications is a very pragmatic discipline: whatever works, regardless of the source, keep it. Due to the changes in technology over the past several centuries, data communications has its origins in many sources. Much of the jargon associated with industrial data communications owes its uniqueness to this mixed parentage. This appendix will examine the major sources of technology that have developed into the modern practice called data communications.

To paraphrase the philosopher George Santayana, (not to be confused with the contemporary Carlos, a rock musician), "those who do not study history are doomed to repeat it." Whether this was originally applied to students in a history class or to world leaders, it is particularly true in data communications. The concepts once tried in one technology keep coming back in the newer versions.

Instrumentation Sources

Automatic control has a form of data communications in the feedback path that is required for closed-loop automatic control to work. Figure B-1 illustrates a feedback control loop.

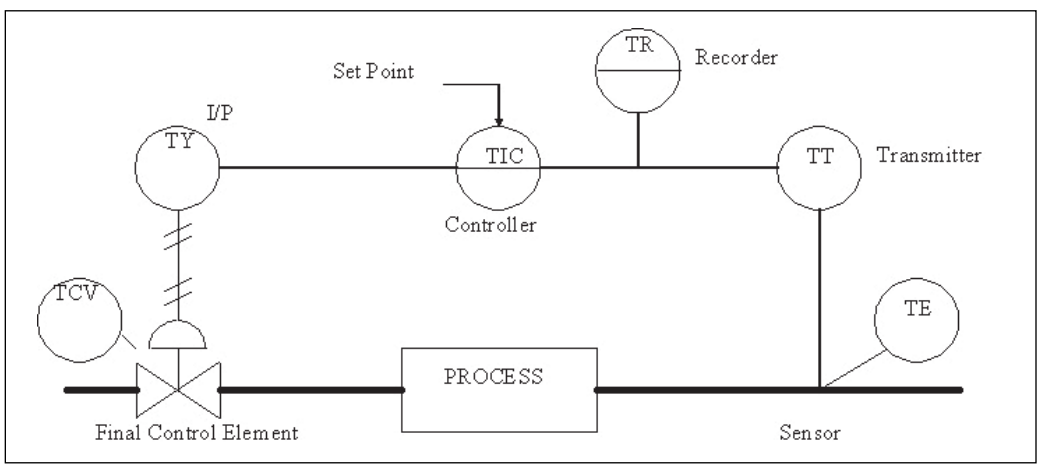


Figure B-1. Feedback Control Loop

For many years, the method of transmitting information to the controller and to the indicators was labeled *signal transmission*.

In the early 1800s, the precursor to many automatic machines was developed: the Jacquard Loom. Its control was a punched card, actually a punched roll like that used on a player piano. The punched card directed the loom weave and as such, is an early form of machine control. More than any other singular concept, the use of a punched card for information storage was to form the basis for the beginnings of data communications a century and a half later. While instrumentation brought technical sophistication to pneumatic signaling, it was the industrial requirements that pneumatics couldn't fill that evolved into today's industrial communications. Pneumatics lack speed. Traveling at the speed of sound is no match for traveling at nearly half the speed of light, as electrical circuits are wont to do. Electric signals are fine for signal transmission, but they lack (for the time being) a cost-effective method of moving large objects like valves, so much of instrumentation was developed around electrical signaling combined with pneumatic activation.

Electrical signal transmission never totally displaced pneumatics in signaling; it took a small piece of silicon to do that. In commercial usage, it would seem one of the most cost-effective implementations of a large computer would be in integrating an entire plant's controls. This was tried in the 1960s; it was called direct digital control (DDC). Figure B-2 is a diagram of a DDC system.

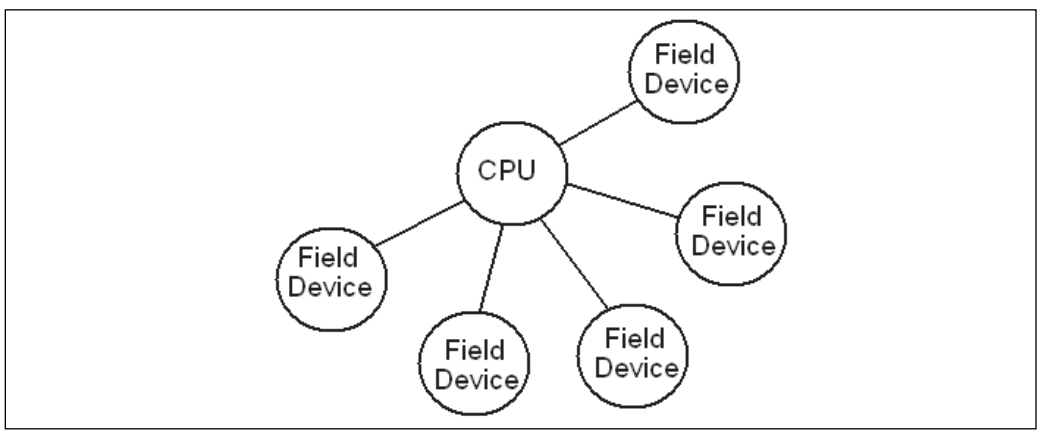


Figure B-2. Diagram of DDC System

These were of a proprietary nature and had several major drawbacks. Note that while the older mainframes had enough computing power to manage a process, they lacked reliability in that time frame. The mean time between failures was not long enough to ensure reliable and error-free performance for any extended period of time, unless, of course, an analog (electric/pneumatic) backup was in place, or enough financial resources were available to permit redundant computers. It wasn't a bad idea, but industrial processes are typically not very tolerant of control failures, particularly if all inputs and outputs are controlled by one point and that point stops working.

One way to avoid this problem is to have the computer control only set points and perform data acquisition for optimization; this is called supervisory control. Then if the main computer goes down, the controllers operate at their last set point and only the optimization provided by the large computer is lost. The supervisory control system uses the same configuration as the DDC system.

Another solution is to use a number of computers with overlapping responsibilities so that if one goes down, the entire process is not taken down, just that part of the process. This is the idea behind *distributed process control*. If the other computers can take up a portion of the down system's functions, so that operation is still possible, it is (in theory) a fault tolerant system.

Note from figure B-3 that an essential part of distributed control systems is the communications between the computers. Just when this evolved into a local area network (LAN) depends upon one's definition and the point in history being looked at. This was a prohibitively expensive scheme in 1960. Even in the 1970s, it would be extremely difficult, from a cost standpoint, to implement. However, the aforementioned little piece of silicon, the microprocessor, allowed this arrangement to become the arrangement of choice since the 1980s. Again note that data communications is essential to the operation of distributed process control by definition.

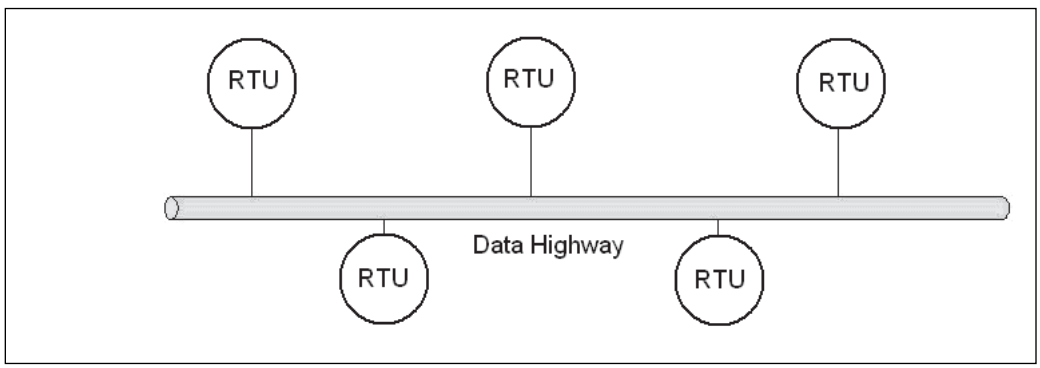


Figure B-3. Distributed Control System Diagram

Telecommunications Sources

One of the primary determinants of modern industrial data communications was the telecommunications industry; the telegraph, and teletypewriter.

Morse Code and the Telegraph

This history, for all practical purposes starts in 1841 with the advent of the first effective binary digital communications system: Samuel Morse and his improved telegraph. While Morse didn't invent the telegraph, what he did invent was the Morse code. Figure B-4 illus-

trates a simple schematic of the Morse-type telegraph system. Morse's machines actually used a mechanical stylus not a sounder. Since telegraphs followed the railroad (right of way) they were networked. Operators noticed that they could tell when their station was being addressed by the sound of the stylus, so sounders were used instead of styluses. Sounders were used because it is easier for the operator to translate the audible signal to the written word rather than from a ticker tape to the written word.

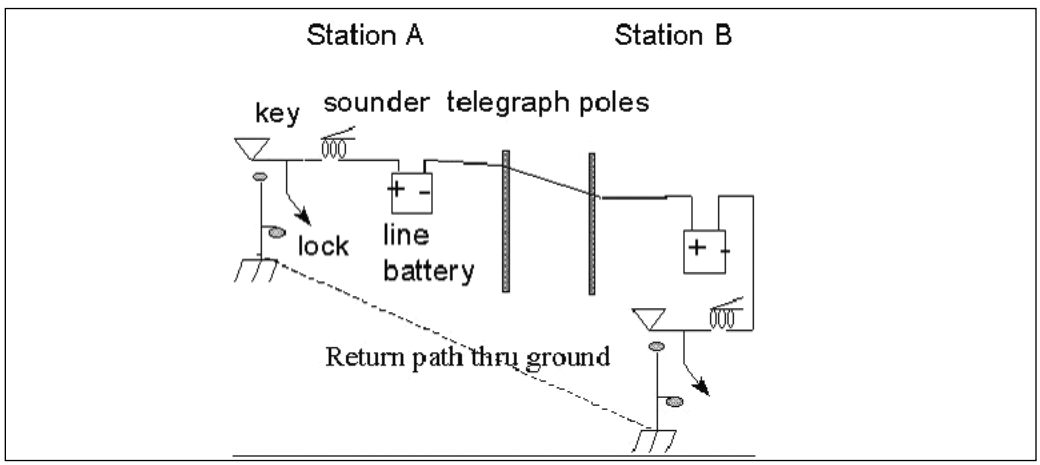


Figure B-4. Simplified Telegraph System

The practical telegraph had significant features that influenced many years of telecommunications design. Note that, since it is a series circuit. Station A has its lock closed as does Station B. This is the protocol used when neither station has any data to transmit. Looking at the block diagram shows, that with both locks closed, there will be current on the line. This is necessary so the receiving end of the communications will be aware of the sending end's desire to transmit. To transmit, the transmitting station will open its key lock. This puts a space (no current) condition on the line, opening the sounder at the distant end. This indicates a "request to send."

This is a *half-duplex* circuit: either end may transmit but not simultaneously. If there had been only a receiver at the distant end without any transmit capability, it would be known as a *unidirectional* (or "pony") circuit in which transmission is possible in only one direction.

Note, too, that if Station A were sending to Station B and Station B developed a high-priority need to send, a simple "protocol" was used. A protocol is nothing more than an agreed-upon set of actions for a set of conditions. To interrupt Station A, Station B opened his lock. This caused the line to go dead, just as if there had been a break in the line itself. When this happened Station A knew through the protocol that he was to stop sending, put his lock on, and wait to receive Station B's message, that is, of course, if this were an intended "line break" and not a real one somewhere between the two stations. This term has been

shortened over the years to *break*. In fact a look at an utterly modern computer keyboard, will, in most cases reveal a key labeled "break." It is there for the same reason as the telegraph line break. Today however, it must be recognized by software in order to perform its protocol steps.

Morse's greatest contribution wasn't in hardware, however; it was in his code. A modern version of Morse code is not much changed from the original version, which is illustrated in figure B-5. The letters most frequently used in the English language have shorter code combinations. Hence, "E" (the most often used letter) is simply a short current pulse. This consideration of letter frequencies was a factor in older codes.

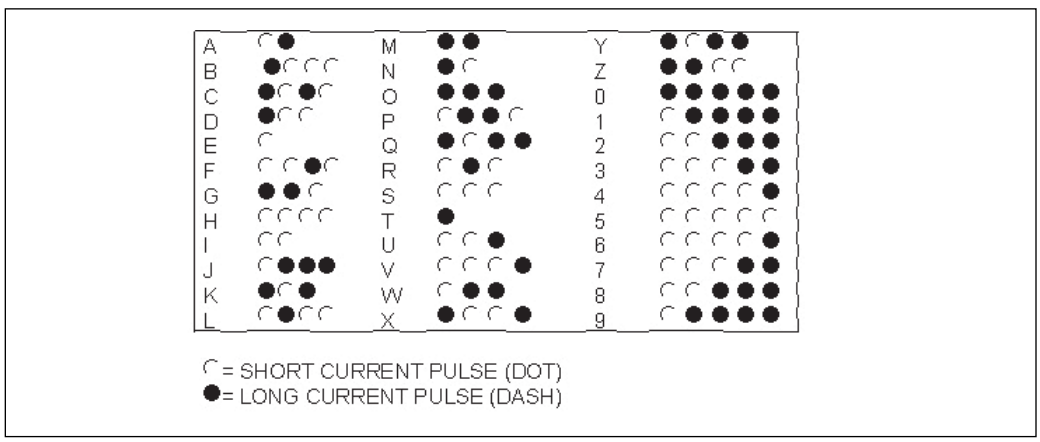


Figure B-5. Morse Code

This is an early form of "data compression." Signaling devices have used many methods other than sounders. Morse's original setup used a device similar to a "ticker tape," An electromagnet pulls a (non-writing) stylus onto a ribbon of paper to make a mark. A "no current" condition leaves a space. This is the most likely origin of the terms "Mark" and "Space," which have been used for many years in the teletypewriter field and are still found in modem literature.

The Teletypewriter

In the late 1800s a new office machine made its debut: the typewriter. Although the first versions bear little resemblance to modern versions, they served the same purpose in that they freed an office from calligraphy requirements. Until that time, good handwriting was, of course, a valued asset. With the advent of the typewriter, more women begin to enter the office administrative fields a province once held almost exclusively by men.

As it was a popular device almost instantly, it wasn't long before many people were thinking of ways to apply the typewriter at a distance, that is, the "teletypewriter." This device continued the trend toward communications networks started by the telegraph. The

telegraph followed the right-of-way of the original mechanized networks - the railroads. The teletypewriter has contributed in great extent, the technological foundations of data communications.

Many designs were considered. Basically, the device had to take a keystroke and transmit information about that keystroke to a distant device that would print the keystroke. A few designs considered using a common plus 26 or more lines—one for each letter. That, of course, was not cost effective for the application, but it is an example of "parallel" transmission. In *parallel transmission*, a single letter is a character and all character information is sent at once. To use a single pair of conductors to transmit data, a character would have to be coded into a series of on and off pulses. This is known as serial transmission, where one element follows another. To consider the magnitude of the technical problem, it would be wise to recall the technical environment of the time.

While the telegraph was already established, the telephone (as a form of telecommunications) was rudimentary, although "networks" of these speaking devices were appearing with greater frequency. The telegraph was the most complex network, in that it followed the great railroad networks—literally.

Switching and routing, a necessary function for railroads, were performed by protocols and routing instructions included in the preamble of a telegraph message. A teletypewriter network would then be patterned after the telegraph since it was more advanced than the telephone.

There were no amplifying devices and no way to perform electronic synchronizing, so mechanical means were required to synchronize two connected teletypewriters for both the character timing and motor speeds. For character timing there had to be a code, some way to convert the parallel action of a keystroke into a series pattern of on and off that would be recognized at the opposite end.

ITA #2 International Telegraphic Alphabet #2, is commonly known as the Baudot code, although sometimes it is called by other names depending upon minor variations. The code was meant primarily for text transmission. It has only uppercase letters, and was quite usable with paper tape as the storage medium. Most early teletypewriters used basically the same circuit as the telegraph with the mechanics of the typewriter. As with the telegraph, the teletypewriter at one end had to have some way of knowing when the other end wished to transmit, so a mark signal (current) would be sent as a *line idle* indication. Since a mark is the idle condition, the first element (now days called a bit) of any code would have to be a space (no current) signal; indeed it is known as the *start space*. Also, the current on condition (mark) needs to exist after the character code pulses have been sent, so the receive device will know when the character is complete, and separate this character from the next character to be transmitted.

This period of current is known as the *stop mark* and is either 1, 1.42, or 2 elements in duration. An example of a teletypewriter signal is illustrated in figure B-6, along with the teletypewriter code (ITA #2) in figure B-7.

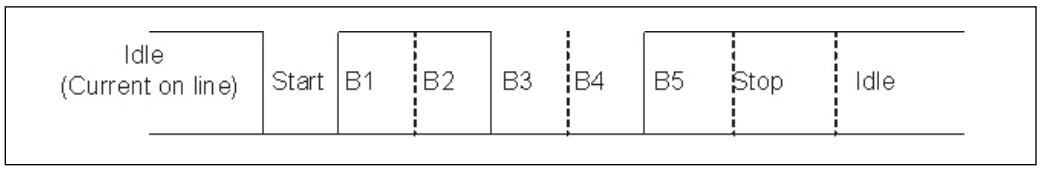


Figure B-6. Teletypewriter Signal

The element (bit time) duration is determined by motor speed. How many elements are needed to encode the English language? There are 26 letters, 10 numerals, and various items of punctuation. Also, since this was an electric typewriter, so to speak, some code patterns must return the carriage, advance the paper, and add spaces between words. While this requires more than 32 patterns (2 to the 5th power), or 5 elements, the developers had help. They were transmitting only the uppercase letters, and they had the typewriter as a model. Therefore, they put in a mechanical shift and used 5 elements: 26 patterns for letters and 26 patterns shifted for numbers and punctuation. Only 26 were available in either shift because 6 patterns were the same for both: Carriage Return, Line Feed, Space, Shift Up, and Shift Down. (If you have added correctly, you might wonder what happened to the one missed. Remember, this is a current-activated, electromechanical device. There is one combination that cannot pull in a clutch magnet and that is the all space [no current] combination. It is known as a *blank*.)

Bit Arrangement							Bit Arrangement						
1	2	3	4	5	Letter	Figure	1	2	3	4	5	Letter	Figure
1	1	0	0	0	A	-	0	0	1	1	0	N	,
1	0	0	1	1	B	?	0	0	0	1	1	O	9
0	1	1	1	0	C	:	0	1	1	0	1	P	0
1	0	0	1	0	D	$	1	1	1	0	1	Q	1
1	0	0	0	0	E	3	0	1	0	1	0	R	4
1	0	1	1	0	F	!	1	0	1	0	0	S	Bell
0	1	0	1	1	G	&	0	0	0	0	1	T	5
0	0	1	0	1	H	@	1	1	1	0	0	U	7
0	1	1	0	0	I	8	0	1	1	1	1	V	;
1	1	0	1	0	J	'	1	1	0	0	1	W	2
1	1	1	1	0	K	(	1	0	1	1	1	X	/
0	1	0	0	1	L	)	1	0	1	0	1	Y	6
0	0	1	1	1	M	.	1	0	0	0	1	Z	"
0	0	0	1	0	Carriage Return		1	1	1	1	1	Shift Letters	
0	1	0	0	0	Line Feed		0	0	1	0	0	Space	
1	1	0	1	1	Shift Figures		0	0	0	0	0	Blank or Null	

Figure B-7. ITA#2 Coding

This code is still the most efficient transmission code for narrative text (in terms of transmission overhead) because it requires very little in machine operations or error detection. While no

longer widely used, at one time it was the most extensively used digital transmission coding. Why the varying stop elements (like 1.42)? To allow the receiver time to finish printing the received character before the next character was transmitted. It would be more correct to call each element a binary digit, or "bit" for short. Even today various devices must be set for the correct number of data and stop bits, although the stop bits are whole units now, either 1 or 2.

When punching a paper tape (until the 1970s it was the most widely used storage medium) if one made an error, it was only necessary to back up the tape to the errored character and over-punch it with the LETTERS key. This made it all ones (all five holes were punched in the paper tape). The only effect this would have on the receiving machine was to put it in the lower or LETTERS position. It was highly probable that it was already in that condition. If in the figures condition, it was a simple matter to punch the next character as FIGURES and go on his or her way. Note that letter frequencies affect this code, with the most frequently used letters having the least number of elements to eliminate wear on the machine.

To send a teletypewriter message on a network, it had to be formatted correctly. Generally a message would have an identifier, then routing information, then the message. By keeping a message count and requiring acknowledgments, a primitive form of error detection was performed, with error correction accomplished by retransmitting the message. Needless to say, many of the teletypewriter conventions have found their way into modern day data communications. This has come about, even though more significant telecommunications developments occurred in other fields and the teletypewriter changed little, using the same code and basically the same equipment until recently.

Telephony

From the early 1920s onward, the telephone developed, radio came into being, and radio networks were established, the radio networks usually using both teletypewriter and telephone links as the connection between stations. Prior to the advent of World War II, the dial telephone network had largely replaced operators in urban areas, but the long distance network had not yet been fully automated.

Various forms of frequency division multiplexing were used on the long lines. *Multiplexing* is the term used to represent placing multiple signals on one line. The signals can be separated by frequency or time. Frequency division separates the different signals by frequency, much the same way you tune in different channels on your television - in fact cable TV is an excellent example of *frequency division multiplexing* (FDM). Long-distance trunks used FDM. In railroad talk, a trunk line is one which carries a lot of traffic, usually between switching centers. The signals on a telephone trunk line may have many different originations and destinations, but are carried on this common pathway.

Another method used sparingly, because its costs per channel greatly exceeded FDM until

the late 1960s, was *time division multiplexing* or TDM. In this system, each sample is assigned a slot in time. This means the slots can be dropped out anywhere a station is in synchrony with the signal. This has evolved into the sophisticated systems used on long lines today. And as the price of complex electronics steadily dropped, the employment of these systems became more widespread. Anyone making a long distance call in the North American continent today has had their voice digitized, multiplexed, de-multiplexed and re-converted to an analog signal. FDM circuits are no longer used on long distance lines, only all-digital signaling. In fact, in the U.S. the only place the telephone is analog is from the central office to the subscriber, the "last mile" so to speak.

Television

One of the technologies that changed both technology and culture is television. A modern television set (and its counterpart the VCR) is quite probably the most technically sophisticated device you will ever own. If memory serves the author correctly, in 1956 a color TV (26" diameter round tube) cost about $525. It had between 19 and 25 tubes, and a minimum of 9 controls on the front, which you had to adjust each time you changed the channel (for those of you not nearly born in 1956, this would be the channel selector, fine tune, volume, horizontal hold, vertical hold, brightness, contrast, color intensity, and tint). Nowadays, a 25" rectangular color console TV bought in a furniture store (not discount house) costs about $525 and generally has one control - the remote. If the set flips one time it is generally assumed that it is time for a new one. The current TV set is absolutely better than its 1956 predecessor in all respects. Yet that dollar in 1996 is worth about 1/10 of that 1956 dollar. Electronics is one area where the performance has increased while the true costs have decreased.

Cable television is a broadband network (in fact a broadband LAN generally uses most of the cable TV parts, right down to the connector). Visual effects demonstrated by television and the ease in conveying information visually has not been lost in either the computer or the automation worlds. Graphical User Interfaces (GUI) have been and are the design goal, and "visually apparent" is a battle cry in manufacturing systems. The effects of television (not just its technology, but its "NOW!" immediacy) have profoundly affected manufacturing systems and will continue to do so as long as it is a human effort.

Data Communications

In the 1890 census, a new form of tabulation took place using machines developed by a gentleman named Herman Hollerith. Perhaps he had taken his inspiration from the Jacquard Loom and its punch cards - really a string of cards akin a bit to a piano player type roll - whatever the source of his inspiration, Mr. Hollerith developed a method of punching cards that allowed most characters to be stored using only one or two punches in each column and absolutely no more than three. This avoided the dreaded "wimpy" card (a card with so many holes it lacked structural strength and was therefore subject to all forms of physical disaster). He was able to successfully store census data on these cards and tabulate

this data mechanically.
Mr. Hollerith went on (aided by the efforts of several others) to form the International Business Machines (IBM™) Corporation. IBM stored business data on (of course) punched cards. Hollerith punched card circa 1947 is illustrated in figure B-8.

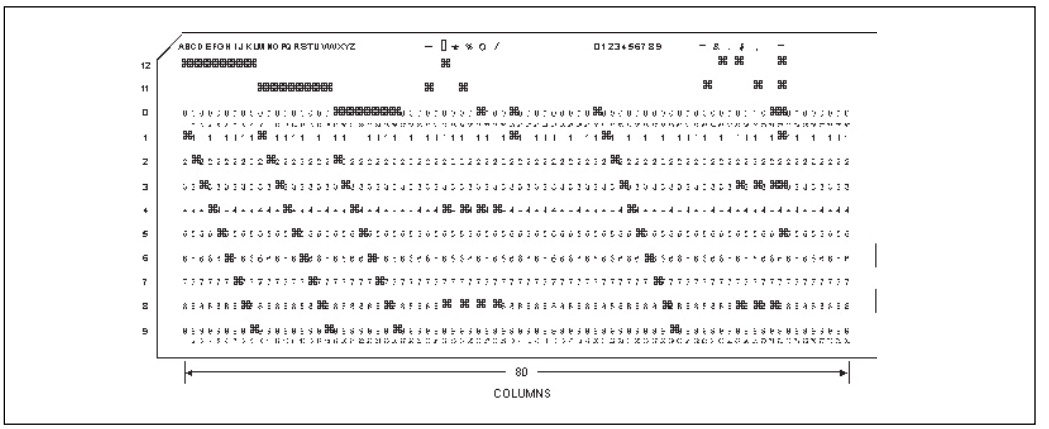


Figure B-8. IBM Data Card

The main method of data transmission at that time was the U.S. mail. After World War II, companies looked for methods of electronically transmitting the card data. The Hollerith Card Code, which had 12 elements, would not be efficient.

However, there was a problem in using the teletypewriter code: while quite efficient, it had no means, other than human, of detecting errors. Inventory codes do not have context as a narrative text would; therefore, human determination of error was out. IBM settled on a "4 of 8" code in which four of the total 8 bits had to be "1"s.

Data communications moved from transporting information over the Wide Area Network (WAN) to Local Area Networks (LAN). In the process, many of the fundamental applications, technologies, and approaches were adopted by the manufacturers of instrumentation, primarily Distributed Control System (DCS) and later Programmable Controller (PLC) manufacturer's "data highways". These areas are indeed the reason for this text's conception. Data communications terminology and techniques have become widespread throughout instrumentation applications, indeed all forms of automated manufacturing.

Computer Technology

Data communications really started between mainframes. A *mainframe* was (and is) a large computer, whose performance is such that it may handle many tasks simultaneously (at least appear to do so). They have always been large physically (in comparison to lesser performing contemporaries), near state-of-the-art (for a current inventory model), and expensive. They were a centralized approach to computing. While making things comfort-

able for the administrative types (upgrades, security, and control), they also were a single point of failure. In the early days, computers were down as much as they were up, due to the technology of the times, so they also constituted a single point of frustration. Mainframes were the only real computers around for the first decade of computing (1947 through 1957). Limited performance, lower cost computers, known as minicomputers came into being and held sway as the lower cost alternative until the middle 1970s.

Minicomputers gave birth to the computerization of process control. They were really smaller versions of the mainframe and usually adhered to the star topology with the computer as the central device. When connected to peripherals or even another computer, they tended to use proprietary architectures and codes.

Enter the microprocessor. First thought of by serious computer types and most Information Systems managers as a "toy," it quickly (ten years is quick) gained respectability. The price/performance ratio has so turned that many large computer systems are now sophisticated arrangements of microcomputers. As it became cost-effective to put micros at every controller and indeed at every part of an instrument loop, it became painfully obvious that they had to communicate in order to control. Methods used in the past were first applied, and those methods have evolved into our modern industrial data networks.

Current Status

The basis and rationale behind most systems comes from experience, not only in the three previously described technical areas, but in many other areas which have contributed this point or that technique. But what is the necessity to connect our "islands of automation" with other administrative and management systems? Competitiveness, economic constraints, proper management of resources, better control of the process, customer directed manufacturing...all these reasons and more. Production does not stand by itself. In any manufacturing company, there are some common threads. Orders must be taken, the resources to fabricate or produce the order must be procured, the correct operational staff (maintenance, too) must be in place as well as the correct process. The manufactured item must then be shipped and, most importantly, the customer billed. It is extremely important that on-time shipments be made to the customer, that no excess inventory or work in process accrues, and that no time in the schedule is wasted in waiting for this resource or that. Small lot sizes and short cycle times are a hallmark of the "world class manufacturer." Small lot sizes usually means the loss of the economy of scale (manufacturing 100,000 widgets will provide more discounts on supplies and materials than 50 widgets). Short cycle times mean frequent tool or recipe changes resulting in a higher ratio of overhead to production. Without industrial data communications, it would not be possible to meet these conflicting requirements and still be competitive and profitable.

Add in the Internet, which provides fast, inexpensive, digital communications; and it becomes possible to tie all of a company's departments together, around a "web site" that customers and company staff utilize for ordering, shipping, and all the functions a business must perform.

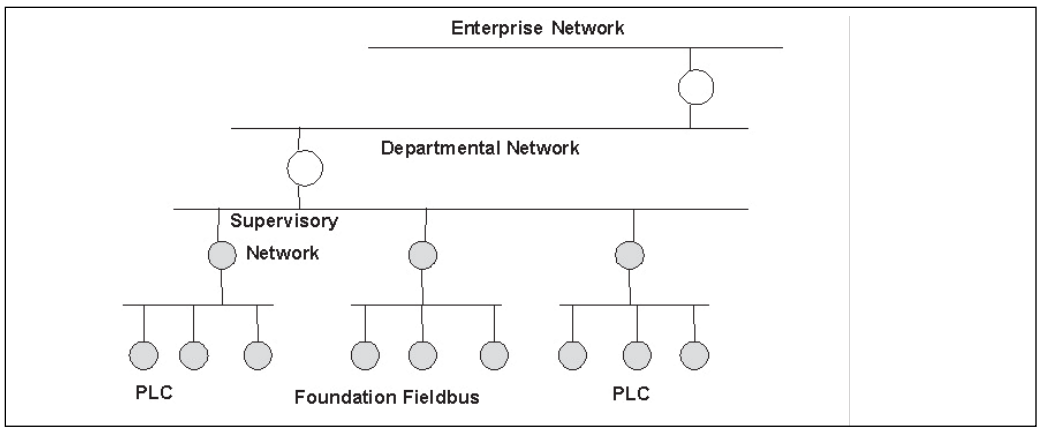


Figure B-9. Hierarchy of Systems

The current status in industry is that it is proceeding toward the goal of relatively open systems, non-proprietary network standards, interconnection between disparate devices, and the inclusion of control and measurement systems in a master integrated network. The goals are closer than ever to a reality today, yet the implementation will still take time, as vendors and users sort out which are viable systems, which implementations are cost effective, and which systems meet their needs. Figure B-9 illustrates a block diagram view of such an integrated system.

Bibliography

Many varied and unacknowledged (perhaps unknown) sources over the past 40 years of instruction in, and the practice of installation, maintenance, and repair of teletypewriter, data communication, and telecommunications devices. The author has observed telecommunications and data communications for the last half century and both lived and worked in this history.

Appendix C
Media

In the text we look at the way data is represented (a set of patterns referred to as "1" or "0") generally in a set of these patterns called an octet or a byte (or sometimes a character when referring to text), and we have looked at the abstraction of a model to see the functions required to move these data representations from one end user to another. We will now look at another of the fundamentals of communications, the medium over which we transmit those data representations, the very bottom of the OSI Model, the media. This is the location where the Physical layer attaches to the outside world, the conduit for information.

All things in data communications are related, and if they are all not correct you do not communicate very well. Sometimes called pipes, the media is the main constraint on data transfer speed. There are three types of media in general we will discuss, these are: the copper types, fiber-optic types, and wireless types. Since this probably covers all media normally used, that isn't too restrictive. Of the copper types, three generally come to mind in the LAN and industrial areas: the Unshielded Twisted Pair (UTP), the Shielded Twisted Pair (STP), and coaxial cable.

UTP

Unshielded Twisted Pair is just as the name implies. It is two conductors twisted over each other. In theory, by twisting the two, an induced field (such as spark noise or other electromagnetic radiation) will be induced equally in both conductors, and since they are of opposite polarity in relation to each other (the signal voltage is across the pairs, not across one conductor and the 0 DC reference known as ground) they will cancel out. This, of course, is dependent upon the number of turns per inch and the uniformity of the twist symmetry. In other words, the larger number of turns per inch, and the more uniform the turns, the higher data rate the cable can pass and the typically, the higher its cost. Normally there is either two or four pair (4 or 8 conductors) surrounded by a sheath that keeps them all together. They are color coded. That is one cable will be a solid color, its partner will be (usually) white with a trace of the partners cable color, typically a stripe so it may be identified. According to EIA TSB 36 (a cabling standard) UTP has 6 Category Levels, 1 through 6, with 1 being the lowest performer and loosest standard to 6 which is the tightest standard and passes the highest data rate.

Category Level	Typical Data Rate	Some Applications
1	.3 to 3 KHz	Telephone lines
2	4 MBps	4-MBps Token Ring
3	10 MBps	10BaseT Ethernet (802.3)
4	16 MBps	16-MBps Token Ring
5	100 MBps	100BaseT Ethernet
5e	100/1000 Mbps 100/1000Base-T Ethernet	(100-MHz cable)
6	1000 Mbps	1000BaseT Ethernet (250-MHz cable
6a	10 GBps 10-GBps Ethernet	(500-MHz cable)

Figure C-1. EIA TSB 36 Category Levels for UTP

UTP has a characteristic impedance (theoretically the line exhibits this impedance regardless of length at the frequency of interest) of 100 ohms. It is necessary because maximum power is transferred between matched impedance's. UTP has 100 ohms. So if the source (generator) has an output impedance of 100 ohms, and the receiver (load) has an input impedance of 100 ohms, and you connect them together using UTP there should be no wasted (in heat or reflected power) energy.

Due to the fact that now (2007) you can purchase Category 6 cable substantially less than Category Level 3, it makes absolutely no sense not to just purchase the Category 6 cable (or 5E which was a kind of interim 6), even though you may not be using its capability in the near future. The cables come in two varieties: plenum and non-plenum. The difference is in the insulation materials. The plenum type will not exude toxic smoke when burning, therefore most local code agencies require (as does the National Electric Code (NEC) the use of plenum type cable in crawl and air spaces.

EIA/TIA 568B Wiring for UTP

The EIA 568 Commercial Building Telecommunications Cabling Standard specifies which color pairs should be used when wiring RJ-45 jacks and plugs. For 100 Mbps and above it is essential that Pairs 1 and 2 be reversed in relation to each other. There are two ways of doing this, 568A and 568B, as shown in figure C-2. Both work fine for direct cables as long as both ends are wired the same. If one end is 568A and one end is 568B you then have a cross connect cable.

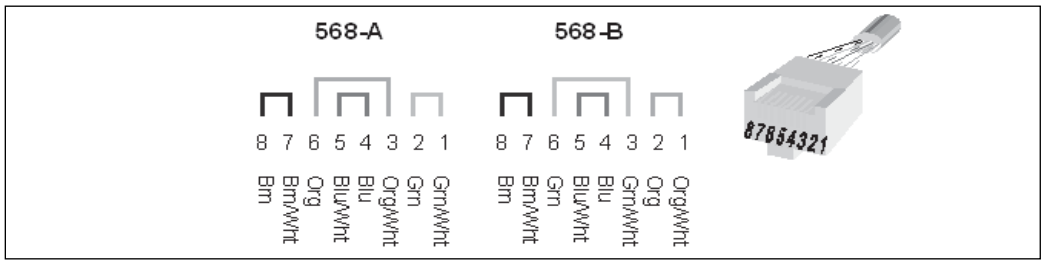


Figure C-2. UTP Wiring

Shielded Twisted Pair

This is the type of cable most persons with any industrial electrical experience know as signal cable. It has one or more twisted pairs enclosed within a metallic sheath called a shield. In turn this sheath is enclosed in a non-conducting sheath. The size of the conductors, whether each pair has a shield or all pairs just have an overall shield, or all pairs are individually shielded as well as an overall shield, and the type of shield (foil or braid) determine the quality and suitability of the conductor. Higher data rates could be passed in theory over STP due to decreased noise levels; however, the shield provides a higher capacitance to ground. Therefore you can pass 1000 MBps for 100 meters using UTP Cat Level 6 and while Token Ring LANs use STP, and have standards similar to those for UTP.

Coaxial Cable

Coaxial Cable, known as coax, is two concentric conductors (see figure C-3) separated by a uniform (physical dimensions and dielectric consistency) insulator, and sheathed in an insulating cover. Coax has for years been used to pass large bandwidths (high frequencies). The reason it may do so is that its characteristic impedance is very closely controlled due in large part to its physical construction. Table C-1 lists some of what used to be commonly used coax types.

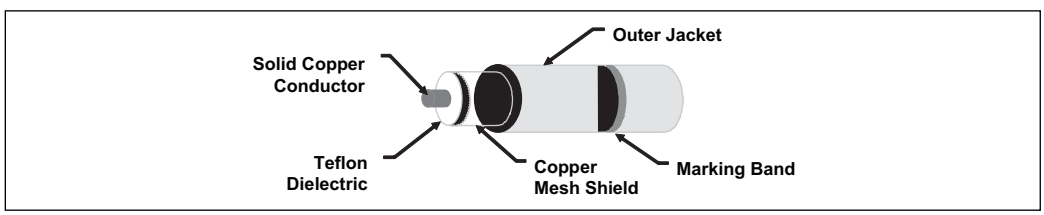


Figure C-3. Coaxial Cable Construction

Type	Impedancee	Comments
RG-8A/U	50	Replaced by RG-212/U
RG-58/U	50	Replaced by RG-213/U
RG-59B/U	75	
RG-62B/U	93	
RG-148/U	72	RG-8/U with spiral armor

Table C-1. Some Commonly Used Types

Good engineering practice (based on many observations as well as theoretical proofs) states you should never use the signal return as a shield. So do not consider a coax as just a big shielded conductor—it is not. The outer conductor is just that, the outer conductor. If you need to shield a coaxial cable, use triax or shielded coax. Coax is more expensive than twisted pair. In the larger diameters it is difficult to install and work with. It is just not as physically flexible as UTP. The higher the quality coax, the less the signal drop per unit length. The larger the coax, the less the signal drop per unit length.

As you may note from table C-1, there are different impedances for coax, the two most common being 50 and 75 ohms. Design of a system with specific characteristic impedance has a lot to do with transmission line theory—a very fascinating subject, but well beyond the reach and scope of this book. As far as we are concerned here, if the system calls for 50 ohm coax, then that is what it gets.

Fiber-Optic Media

Transmission of a signal by using light is an astoundingly simple concept, and since light is immune to all but the strongest electrical environment interference, it would be ideal in industrial areas. If this is so why hasn't it excluded all other media? It is for the reason that simple concepts are sometimes difficult and/or expensive to implement. We will do a cursory explanation of fiber-optic transmission in order to explain the difficulties with the media itself, it should be stated that the high costs formerly associated with fiber-optic transmission have more to do with the interface electronics—how we connect our electronic devices to the cable and the high costs of maintenance when necessary. In any case these costs have dropped dramatically in the last two decades so much so that fiber is competitive with copper in many areas.

Fiber-Optic Operation

The principles of physics that cause fiber optics to work are refraction and reflection. Refraction occurs because the speed at which light travels varies with the media. As an example, light travels faster in air than in glass. Figure C-4 illustrates the principles involved.

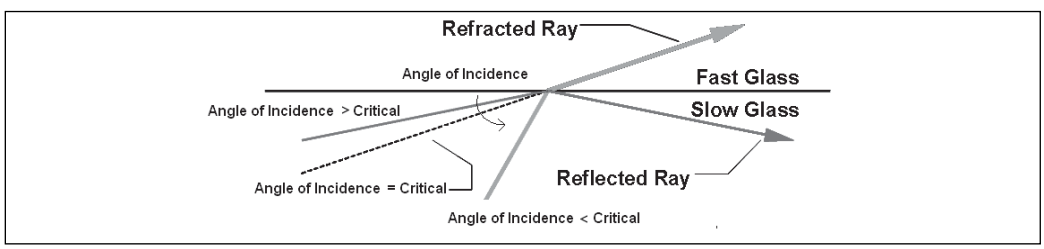


Figure C-4. Incidence, Refraction, Reflection

Note that there is a critical angle. As light approaches the surface of the interface between the fast glass and the slow glass at less than the critical angle, light leaves the glass but is

refracted. At the critical angle, the light never leaves the surface of the interface but travels parallel to it. Above the critical angle, there is total reflection.

The angle of incidence and the angle of refraction are related. The angle of incidence for the critical angle results in an angle of refraction differing by 90 degrees. The critical angle varies for different interfaces. The relationship between the angle of refraction and the angle of incidence is the ratio of their sine's. The "index of refraction" is expressed as

$$\eta = \text{sine (angle of incidence)} / \text{sine (angle of refraction)}$$

This is one of the more important parameters behind the operation of a fiber-optic cable. Since the critical angle is the point between refraction and reflection, all light approaching a fiber-optic cable must be at some angle greater than the critical angle.

Figure C-5 shows the makeup of a fiber-optic cable.

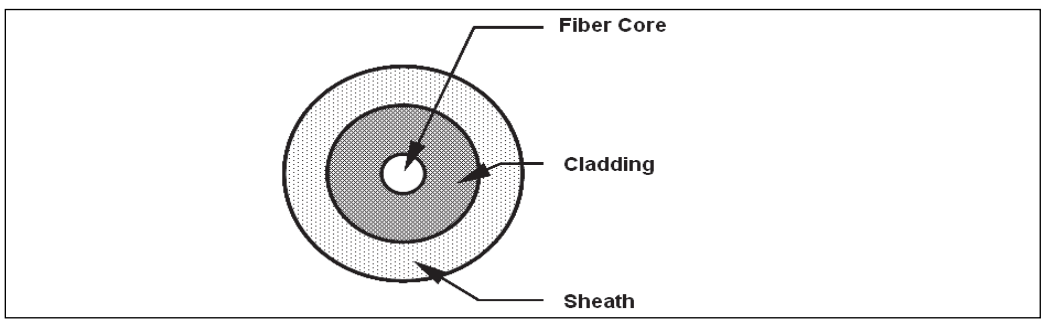


Figure C-5. Fiber-Optic Cable

The cladding and the core material are such that the index of refraction for the core material is greater than the index of refraction for the cladding. Since light must approach at a greater than critical angle, if a line is drawn that extends from the center of the fiber-optic cable and another line is drawn from the surface at the critical angle, the angle of acceptance is produced. If this angle is rotated a cone is generated, called the cone of acceptance. Light must fall within the cone to be transmitted through the fiber. Figure C-6 illustrates the "cone of acceptance."

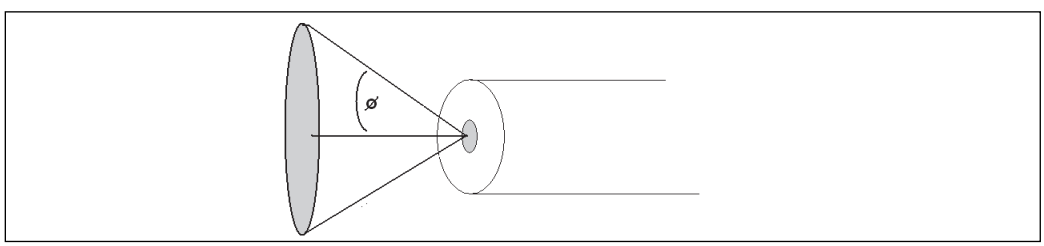


Figure C-6. Cone of Acceptance

Losses

Fiber optics have some particular losses. One is scattering, another is absorption. Scattering is the act of light being spread out by the molecular structure of the fiber material. Scattering is also caused by impurities in the fiber, bubbles, scratches, and any imperfection in the fiber surface.

Manufacturing techniques and good installation practices can eliminate all of the scattering losses except that caused by molecular structure. This is a fundamental loss that cannot be reduced further. Absorption losses are also caused by impurities in the fiber material. An almost semiconductor purity must be obtained if a fiber is to limit absorption loss.

The size of a fiber-optic cable determines its mode of operation and its losses. Single mode fibers have a very small diameter fiber in relation to the cladding. Multi-mode fibers have a larger ratio core-to-cladding size ratio, and plastic fibers have a core of higher index material than the surrounding material. Plastic fibers are multi-mode and are used for relatively short distances and only at restricted temperatures. Single-mode and multi-mode refer to the number of paths and hops light can take to reach the end (actually they are much more similar to the modes of a wave guide but we already eliminated transmission line theory from consideration). Single-mode fibers have good transmission throughput and are easier to handle, but require far more precise mechanical alignment than the multi-mode types (also they cost more). Figure C-7 is an illustration of some fiber-optic sizes.

Mode	Core mm	Cladding mm
Single Mode	8	125
Multi-mode	50	125
Multi-mode	62.5	125
Multi-mode	100	140

Figure C-7. Selected Fiber-Optic Sizes

Fiber optics would appear to be the answer to most industrial applications. That they have not been was due to high cost, not necessarily of the fiber-optic material but to the electronics for getting on and off of the cable. However, due to fiber's capability of much higher bandwidth along with noise immunity and electrical isolation and to the significant reduction in cost in the last few years, fiber has come to be the recommended media for LAN backbone cables (the main trunk cable hosting many networks) by many automation vendors. History has taught where there is a want (or need) eventually the answer will be

cost effective. Look for fiber-optic media to continue to make inroads into industrial areas for interconnection; however, it is restricted when tying field devices together by its inability to supply power to the device using the signal media as in the present two-wire loop.

Wireless Media

Actually to talk about the wireless media quite properly we should be talking about the atmosphere, which is where the energy is propagated. Wireless is just that, radio communications. But this is not a simple amplitude modulated (AM) signal or even an FM broadcast type signal. When referring to wireless, we are referring to a method of propagating signals through the air (or vacuum) by impressing them on an extremely high frequency carrier signal (for industrial applications it is UHF - ultra high frequency, between 800 MHz and 5.2 GHz).

Wireless has been around for some time but was not used in industry for a number of reasons, the primary one being the high ambient electrical noise in an industrial facility. Other interference factors from external sources such as walkie-talkies, fixed radio and TV, plus multi-path reception have given this method a less than reliable reputation in such an environment.

A signaling method designed to overcome jamming is proving effective in the industrial area. It is known as *spread spectrum*. In the frequency-hopping type (FHSS) a predetermined algorithm selects the transmit frequencies throughout the authorized spectrum, never spending more than a packet or so at any one frequency, the receiver has the same algorithm so tracks along with the transmitter. Another method is *Direct Sequence Spread Spectrum* (DSSS) – in either case the signal is spread over a wide band of frequencies and the algorithm used whether hopping or spreading is known only within the system. These are two of the methods used in 802.11. The third method used is *Orthogonal frequency-division multiplexing* (OFDM), also called *discrete multitone modulation* (DMT). This is the original technique used in many wireline modems and is now used in wireless modems. Spread spectrum and OFDM are a more efficient use of bandwidth than assigning each person a transmit frequency for all or even a single transmission.

In industrial networks, wireless applications are not at this moment widespread, but the instance of use is rising. When one realizes just how much money is "capital at rest" in a conventional wiring system, and the number of times processes or parts of processes are moved or re-engineered, a wireless alternative may seem like a cost-effective approach. Wireless as an industrial media will have to prove itself in more than support applications before there is widespread installation. Wireless LANS are discussed at some length in chapter 4 of this text.

Media Summary

The three major categories of media used in the industrial area were discussed: copper, fiber optic, and wireless. Of the three, copper is by large the most commonly installed.

Copper has limitations. Both wireless and fiber optic are efforts to overcome these limitations. Each of the alternative to copper methods have limitations of their own which so far have limited their acceptance in the industrial area; however, fiber optic is becoming the standard for transmission of data over distance in a facility or between facilities.

Glossary

abort	to terminate or exit a program or communication hastily without any data or states being saved—not a desirable end to a program
ACK	acknowledgment upon reception of a message or communication
A/D	analog to digital converter
adapter	a device to connect different parts of a system or provide an interface. An example would be an EIA-485 adapter that plugs into a PC and provides 485 termination in place of the PC comm. Port
ADCCP	advanced data communications control protocol—ANSI version of LAP-B (see LAP-B)
address	the identifier for a logical or physical element
agent	a software function that responds to network management requests
AIX	IBM's version of UNIX
algorithm	a set of steps (procedure) to a solution—may be mathematical, but does not have to be
analog	any value between the lower and upper range values—a continuum of values
ANSI	American National Standards Institute
ANSI/EIA-568	commercial building telecommunications wiring standard
API	application programming interfaces standard format for connectivity to a program or function
ARCnet	Datapoint-derived token-passing bus network
ARPANET	a U.S. Department of Defense sponsored packet switching network—the forerunner of the Internet
ARQ	automatic retransmission (repeat) query (reQuest)—an error correction strategy

ASCII	American Standard Code for Information Interchange—a 7-bit character set, usually confused with the 8-bit extended ASCII
ATM	asynchronous transfer mode—a connection-oriented cell oriented high-speed transmission protocol
AUX	Apple Corporation's version of UNIX
AWG	American Wire Gauge—standard for wire sizes
B-channel	Bearer channel in ISDN (see ISDN)
backbone	the main connectivity media for distributed systems
BASIC	beginners all-purpose symbolic instruction code—an interpreted or compiled programming language
baud	a unit of line modulation rate—determined by dividing the time of the smallest occurring element into 1. This is the decision rate the media will support.
BER	Bit error rate—the ratio of errored bits to bits transmitted
binary	a digital system with the base of 2
BIOS	Basic input/output system—firmware that specifies certain results regardless of physical architecture for predetermined inputs
bit rate	bits per second
block	a unit of data usually stored and perhaps transmitted as a unit (see packet and frame)
bridge	a device for connecting two segments of a LAN—contains twin sets of layer 1 and 2 functions
CAN	controller area network
category cable	cable that complies with EIA-TSB-36
CCITT	Comite Consultatif International de Telegraphique et Telephonique—International Telecommunications Standards setting body replaced by ITU-ITS

client/server	arrangement where the client shares server resources and each performs portions of the whole task
CIFS	common Internet file system
CLNP	connectionless network protocol—part of OSI Standards, referred to as OSI-IP
CLNS	connectionless network protocol—one of the two options in the OSI Standards
COM	component object model
CONS	connection-oriented network service—the OSI Standard for connection-oriented service
CRCC	cyclic redundancy check character—an error detection scheme
CSMA/CD	carrier sense multiple access/collision detections media access strategy
CSU	channel service unit—used to terminate a digital line at the customer site
D/A	digital to analog
DBMS	database management system
DCE	data communications equipment—the interface between the DTE and the communications channel
DCS	distributed control system—a networked system of client server devices, where the intelligence is distributed throughout the system to perform parts of the whole task
DDD	direct distance dialing
decimal	a number system with the base 10
DCOM	distributed component object model
DCS	distributed control system
DDS	digital data service
DES	data encryption standards—a cryptographic block algorithm designed by NBS (NIST)

DIX Ethernet	Digital Intel Xerox (also known as Ethernet 11)
DLLs	dynamic link libraries
DSAP	destination service access point—user destination address for a service
DSL	digital subscriber line
DSU	digital service unit—terminates the data circuit to the CSU
DTE	data terminal equipment—equipment or logical location opposed to the DCE, a data processing as opposed communicating equipment
duplex	formerly full duplex—transmission simultaneously in both directions
EBCDIC	extended binary coded decimal interchange code—IBM's proprietary 8-bit character set
end user	the source/destination of data sent through a communications system
EIA	Electronic Industries Association
EIA-232	digital interface up to 20 Kbps, unbalanced to ground
EIA-422	electrical standard for balanced to ground transceiver, up to 10 N4Bps
EIA-423	electrical standard for unbalanced to ground transceiver, up to 20 Kbps
EIA-485	multidrop balanced to ground electrical specification
ES	end system, as defined by OSI
ES-IS	end system to intermediates system, as defined by OSI
Ethernet	Term applied to both IEEE 802.3 and Ethernet 2.0 systems that use CSNIA/CD and differ only in minor structure
FDDI	fiber distributed data interface—a token-passing fiber-optic ring capable of 100 Mbps transmission
FEC	forward error correction—the use of delay and error detection algorithms to protect data and correct all errors within the protection of the algorithm

fiber-optics	a transmission media using light as the carrier
Fieldbus	network connected to field devices; also used for the SP50 Fieldbus, a standard
frame	a sequence of octets with a header and a trailer; usually a Layer 2 contrivance, similar to the Layer 3 and above packet or the older block
frame relay	connectionless packet system
FTAM	file transfer, access control, and management—programs that interface to the applications layer of the OSI Model
FTP	file transfer protocol
gateway	a twin seven-layer device for connectivity between dissimilar systems
GOSIP	Government Open System Interconnection Profile—the U.S. Government version of OSI
GUI	graphical user interface
half duplex	transmission between two points but not simultaneously
HART	highway addressable remote transducers—a method of communicating with smart instruments
HDLC	high level data link control similar to ADCCP and LAP-B
IEEE	Institute of Electrical and Electronic Engineers—a professional society that sets standards
Internet	An international public packet switched network consisting of numbers of backbones and many nodes using TCP/IF
Intranet	corporate networks (Enterprise) connected using Internet technologies
IP	internet protocol—a Layer 3 routing protocol
IPX	Internetwork packet exchange—a modification of IF for LANs
IS-IS	intermediate system to intermediate system, defined by OSI
ISA	ISA—a professional society that is setting the standard for automation
ISDN	integrated services digital network—an older technology for bringing

	digital to the desktop
ISO	International Standards Organization—an international standards-setting organization
jabber	to continuously transmit—indicates a failed adapter
Kbps	kilobits per second
LAN	local area network—more than two nodes connected together serving a function; the media and distribution is privately owned
LAP-B	link access protocol-balanced—a duplex version of a bit-oriented protocol similar to HDLC, SDLC, and ADCCP
LATA	local access and transport area—a monopoly telephone provider service company's area of operations
LLC	logical link control—the upper sub-layer of layer 2 data link
MAC	media access control—the lower sub-layer of layer 2 data link
MAN	IEEE 802.6 metropolitan area network
Mbps	megabits per second
media	plural of medium—the entity used for propagation of the communications signal
MIB	management information base—database used in SNMP
MIS	Management Information Systems—organization providing computing and support
MNP	microcom networking protocols—a set of error detection and compression protocols, MNP-1 through 10
modem	MODulator/DEModulator—converts digital data to analog signals for transmission, then converts back to digital upon reception
multi-tasking	two or more programs segments running simultaneously
multi-threaded	method of segmenting instructions
multi-user	more than one user simultaneously

multiplexor	device for placing more than one signal at a time on a single line
NAK	negative acknowledgment—signal signifying the previous transmission was errored
.NET	a system based in part on XML and SOAP
NetBEUI	NetBIOS enhanced user interface—the NetBIOS Layer 4 transport
NetBIOS	network basic input/output system—a set of commands and protocols to allow network traffic between a workstation and server
NIC	network interface card—the network adapter that plugs into a PC
NIST	National Institute for Standards and Technology—previously, National Bureau of Standards
node	an addressable entity connected to a network
OOP	object-oriented programming
OPC	object linking and embedding (OLE) for Process Control
OSI	open systems interconnection—an open set of standards for networking
packet	a unit of data, usually from Layer 3 or higher (see block and frame)
parity	an error-checking technique dependent upon the number of one states in a character
physical layer	Layer 1 of the OSI Model
PLC	programmable logic controller
polling	a form of master-slave where one station (hub or master) queries all connected stations
RIP	routing information protocols—a method for determining addresses on a multi-server network
SAP	service access point—the interface address between layers
SATA	serial ATA—connects EIDE drives to PC mother boards

SCSI	small computer system interfaces—a method of connecting a peripheral to small systems
SDLC	synchronous data link control—IBM's version of ADCCP, HDLC, and LAP-B
SMTP	simple mail transport protocol
SNMP	simple network management protocol—performs network management over TCP/IP systems
SOAP	simple object access protocol
SSAP	source service access point—the source address of a service between layers
STP	shielded twisted pair—a twisted pair of conductors surrounded by their individual shield
TCP	transmission control protocol—the Layer 4 transport for TCP/IP transmission
TIA	Telecommunications Industries Association
token	a binary pattern
token bus	a bus topology network, where a node must have the token to initiate transmission
token ring	a ring topology network, where a node must have the token to initiate transmission
two-wire loop	an SP50 standard used to connect field devices—supplies power to transmitter and provides signal to receiver
USB	universal serial bus—allows plug-and-play operation for PC peripherals
UTP	unshielded twisted pair
WAN	wide area network—where the media is leased or rented
XML	extensible markup language
zero insertion	in normal transmission, the rule is to insert a zero any time the protocol detects five 1 bits in the data stream

Index