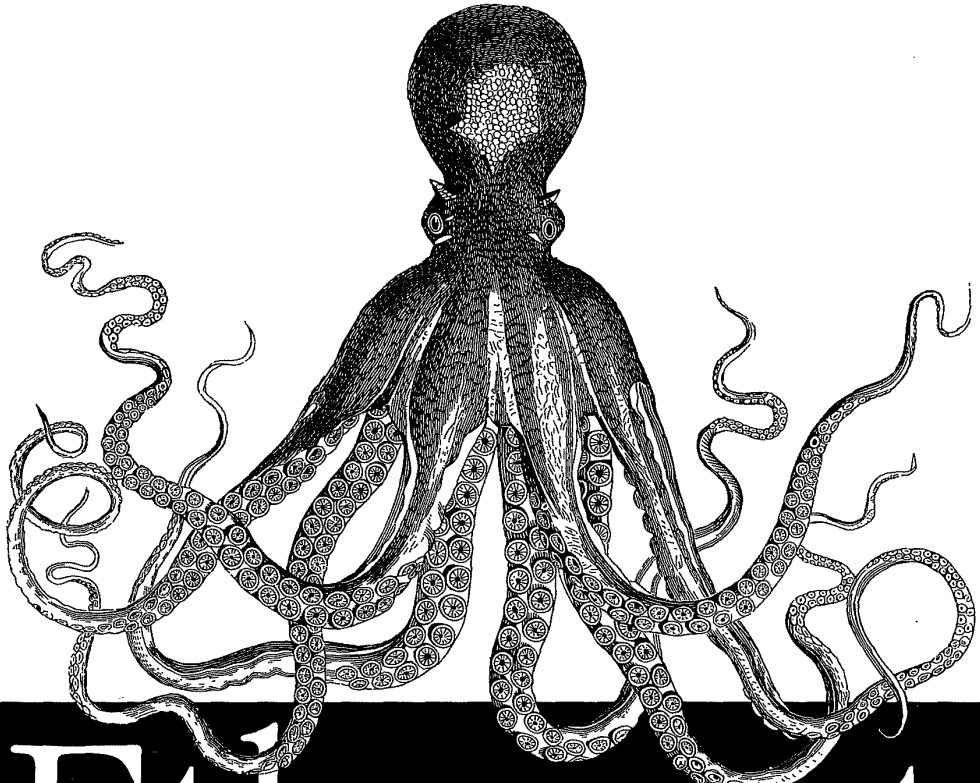
Designing and Managing Local Area Networks



Ethernet

The Definitive Guide

O'REILLY®

Charles E. Spurgeon

Ethernet: The Definitive Guide
by Charles E. Spurgeon

Published by O'Reilly & Associates, Inc., 101 Morris Street, Sebastopol, CA 95472.

Editors: Mark Stone and Chuck Toporek

Production Editor: David Futato

Cover Designer: Hanna Dyer

Printing History:

February 2000: First Edition.

Library of Congress Cataloging-in-Publication Data

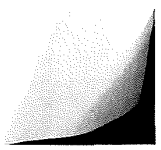
Spurgeon, Charles (Charles E.)
Ethernet: the definitive guide / Charles E. Spurgeon
p. cm.
ISBN 1-56592-660-9 (alk. paper)
1. Ethernet (Local area network system) I. Title.

TK5105.8.E83 S67 2000
004.6'8--dc21

99-086932

[M]

Table of Contents



I

Introduction to Ethernet

The first part of this book provides a tour of basic Ethernet theory and operation. These chapters cover those portions of Ethernet operation that are common to all Ethernet media systems. Common portions include the Ethernet frame, the operation of the media access control system, full-duplex mode, and the Auto-Negotiation protocol.

Part I contains these chapters:

- Chapter 1, *The Evolution of Ethernet*
- Chapter 2, *The Ethernet System*
- Chapter 3, *The Media Access Control Protocol*
- Chapter 4, *Full-Duplex Ethernet*
- Chapter 5, *Auto-Negotiation*

In this chapter:

- *History of Ethernet*
- *The Latest Ethernet Standard*
- *Organization of IEEE Standards*
- *Levels of Compliance*
- *IEEE Identifiers*
- *Reinventing Ethernet*
- *Multi-Gigabit Ethernet*

1

The Evolution of Ethernet

Ethernet is by far the most widely used local area networking (LAN) technology in the world today. Market surveys indicate that hundreds of millions of Ethernet network interface cards (NICs), repeater ports, and switching hub ports have been sold to date, and the market continues to grow. In total, Ethernet outsells all other LAN technologies by a very large margin.

Ethernet reached its 25th birthday in 1998, and has seen many changes as computer technology evolved over the years. Ethernet has been constantly reinvented, evolving new capabilities and in the process growing to become the most popular network technology in the world.

This chapter describes the invention of Ethernet, and the development and organization of the Ethernet standard. Along the way we provide a brief tour of the entire set of Ethernet media systems.

History of Ethernet

On May 22, 1973, Bob Metcalfe (then at the Xerox Palo Alto Research Center, PARC, in California) wrote a memo describing the Ethernet network system he had invented for interconnecting advanced computer workstations, making it possible to send data to one another and to high-speed laser printers. Probably the best-known invention at Xerox PARC was the first personal computer workstation with graphical user interfaces and mouse pointing device, called the Xerox Alto. The PARC inventions also included the first laser printers for personal computers, and, with the creation of Ethernet, the first high-speed LAN technology to link everything together.

This was a remarkable computing environment for the time, since the early 1970s were an era in which computing was dominated by large and very expensive

mainframe computers. Few places could afford to buy and support mainframes, and few people knew how to use them. The inventions at Xerox PARC helped bring about a revolutionary change in the world of computing.

A major part of this revolutionary change in the use of computers has been the use of Ethernet LANs to enable communication among computers. Combined with an explosive increase in the use of information sharing applications such as the World Wide Web, this new model of computing has brought an entire new world of communications technology into existence. These days, sharing information is most often done over an Ethernet; from the smallest office to the largest corporation, from the single schoolroom to the largest university campus, Ethernet is clearly the networking technology of choice.

The Aloha Network

Bob Metcalfe's 1973 Ethernet memo describes a networking system based on an earlier experiment in networking called the Aloha network. The Aloha network began at the University of Hawaii in the late 1960s when Norman Abramson and his colleagues developed a radio network for communication among the Hawaiian Islands. This system was an early experiment in the development of mechanisms for sharing a common communications channel—in this case, a common radio channel.

The Aloha protocol was very simple: an Aloha station could send whenever it liked, and then waited for an acknowledgment. If an acknowledgment wasn't received within a short amount of time, the station assumed that another station had also transmitted simultaneously, causing a *collision* in which the combined transmissions were garbled so that the receiving station did not hear them and did not return an acknowledgment. Upon detecting a collision, both transmitting stations would choose a random backoff time and then retransmit their packets with a good probability of success. However, as traffic increased on the Aloha channel, the collision rate would rapidly increase as well.

Abramson calculated that this system, known as *pure Aloha,* could achieve a maximum channel utilization of about 18 percent due to the rapidly increasing rate of collisions under increasing load. Another system, called *slotted Aloha,* was developed that assigned transmission slots and used a master clock to synchronize transmissions, which increased the maximum utilization of the channel to about 37 percent. In 1995, Abramson received the IEEE's Koji Kobayashi Computers and Communications Award "for development of the concept of the Aloha System, which led to modern local area networks."

Invention of Ethernet

Metcalfe realized that he could improve on the Aloha system of arbitrating access to a shared communications channel. He developed a new system that included a mechanism that detected when a collision occurred (*collision detect*). The system also included "listen before talk," in which stations listened for activity (*carrier sense*) before transmitting, and supported access to a shared channel by multiple stations (*multiple access*). Put all these components together, and you can see why the Ethernet channel access protocol is called Carrier Sense Multiple Access with Collision Detect (CSMA/CD). Metcalfe also developed a more sophisticated back-off algorithm, which, in combination with the CSMA/CD protocol, allowed the Ethernet system to function at up to 100 percent load.

In late 1972, Metcalfe and his Xerox PARC colleagues developed the first experimental Ethernet system to interconnect the Xerox Alto. The experimental Ethernet was used to link Altos to one another, and to servers and laser printers. The signal clock for the experimental Ethernet interface was derived from the Alto's system clock, which resulted in a data transmission rate on the experimental Ethernet of 2.94 Mbps.

Metcalfe's first experimental network was called the *Alto Aloha Network*. In 1973, Metcalfe changed the name to "Ethernet," to make it clear that the system could support any computer—not just Altos—and to point out that his new network mechanisms had evolved well beyond the Aloha system. He chose to base the name on the word "ether" as a way of describing an essential feature of the system: the physical medium (i.e., a cable) carries bits to all stations, much the same way that the old "luminiferous ether" was once thought to propagate electromagnetic waves through space.* Thus, *Ethernet* was born.

In 1976, Metcalfe drew the following diagram (Figure 1-1) "...to present Ethernet for the first time. It was used in his presentation to the National Computer Conference in June of that year. On the drawing are the original terms for describing Ethernet. Since then, other terms have come into usage among Ethernet enthusiasts."†

In July 1976, Bob Metcalfe and David Boggs published their landmark paper "Ethernet: Distributed Packet Switching for Local Computer Networks," in the *Communications of the Association for Computing Machinery* (CACM). In late 1977, Robert M. Metcalfe, David R. Boggs, Charles P. Thacker, and Butler W.

---

* Physicists Michelson and Morley disproved the existence of the ether in 1887, but Metcalfe decided that it was a good name for his new network system that carried signals to all computers.

† From *The Ethernet Sourcebook*, ed. Robyn E. Shotwell (New York: North-Holland, 1985), title page. Diagram reproduced with permission.

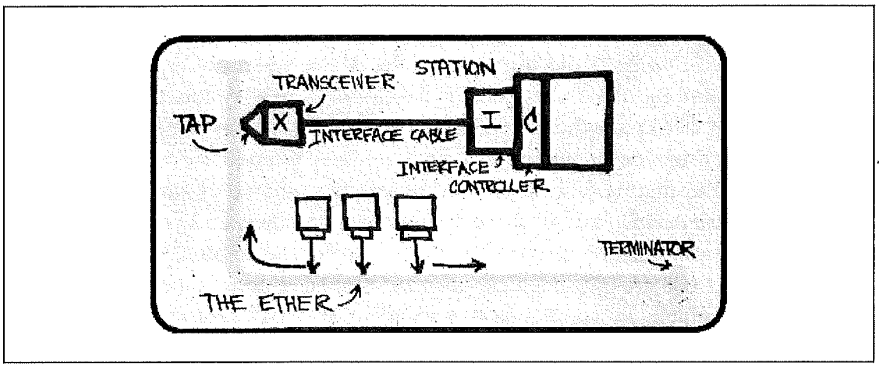


Figure 1-1. Drawing of the original Ethernet system

Lampson received U.S. patent number 4,063,220 on Ethernet for a "Multipoint Data Communication System With Collision Detection." A patent for the Ethernet repeater was issued in mid-1978. At this point, Xerox wholly owned the Ethernet system. The next stage in the evolution of the world's most popular computer network was to liberate Ethernet from the confines of a single corporation and make it a worldwide standard.

Evolution of the Ethernet Standard

The original 10 Mbps Ethernet standard was first published in 1980 by the DEC-Intel-Xerox vendor consortium. Using the first initial of each company, this became known as the DIX Ethernet standard. This standard, entitled *The Ethernet, A Local Area Network: Data Link Layer and Physical Layer Specifications*, contained the specifications for the operation of Ethernet as well as the specs for a single media system based on thick coaxial cable. As is true for most standards, the DIX standard was revised to add some technical changes, corrections, and minor improvements. The last revision of this standard was DIX V2.0.

When the DIX standard was published, a new effort led by the Institute of Electrical and Electronics Engineers (IEEE) to develop open network standards was also getting underway.* Consequently, the thick coaxial variety of Ethernet ended up being standardized twice—first by the DIX consortium and a second time by the IEEE. The IEEE standard was created under the direction of the IEEE Local and Metropolitan Networks (LAN/MAN) Standards Committee, which identifies all the standards it develops with the number 802. There have been a number of net-

* The IEEE is the world's largest technical professional society, with members in 150 countries. The IEEE provides technical publishing, holds conferences, and develops a range of technical standards, including computer and communications standards. The standards developed by the IEEE may also become national and international standards.

working standards published in the 802 branch of the IEEE, including the 802.3* Ethernet and 802.5 Token Ring standards.

The IEEE 802.3 committee took up the network system described in the original DIX standard and used it as the basis for an IEEE standard. The IEEE standard was first published in 1985 with the title *IEEE 802.3 Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications.* The IEEE standard does not use "Ethernet" in the title, even though Xerox relinquished their trademark on the Ethernet name. That's because open standards committees are quite sensitive about using commercial names that might imply endorsement of a particular company. As a result, the IEEE calls this technology *802.3 CSMA/CD* or just *802.3.* However, most people still use the Ethernet name when referring to the network system described in the 802.3 standard.

The IEEE 802.3 standard is the official Ethernet standard. From time to time you may hear of other Ethernet technology "standards" developed by various groups or vendor consortiums. However, if the technology isn't specified within the IEEE 802.3 standard, it isn't an official Ethernet technology. Periodically, the latest IEEE 802.3 standards are presented to the American National Standards Institute (ANSI), which forwards them on, where they are adopted by the International Organization for Standardization (ISO). This organization is described in more detail later in this chapter. Adoption by the ISO means that the IEEE 802.3 Ethernet standard is also a worldwide standard, and that vendors from around the globe can build equipment that will work together on Ethernet systems.

Ethernet Family Tree

The title of the latest version of the IEEE standard as of this writing is *802.3, 1998 Edition Information Technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications.*

This edition contains 1,268 pages and "includes all contents of the 8802-3:1996 Edition, plus IEEE Std 802.3aa-1998, IEEE Std 802.3r-1996, IEEE Std 802.3u-1995, IEEE Std 802.3x&y-1997, and IEEE802.3z-1998." These latter documents were developed as supplements to the standard. This edition of the standard can be purchased from the IEEE through their web site at: *http://standards.ieee.org/catalog/IEEE802.3.html.*

* Pronounced "eight oh two dot three."

The Latest Ethernet Standard

After the publication of the original IEEE 802.3 standard for thick Ethernet, the next development in Ethernet media was the thin coaxial Ethernet variety, inspired by technology first marketed by the 3Com Corporation. When the IEEE 802.3 committee standardized the thin Ethernet technology, they gave it the shorthand identifier of 10BASE2, which is explained later in this chapter.

Following the development of thin coaxial Ethernet came several new media varieties, including the twisted-pair and fiber optic varieties for the 10 Mbps system. Next, the 100 Mbps Fast Ethernet system was developed, which also included several varieties of twisted-pair and fiber optic media systems. Most recently, the Gigabit Ethernet system was developed using both fiber optic and twisted-pair cabling. These systems were all developed as supplements to the IEEE Ethernet standard.

IEEE Supplements

When the Ethernet standard needs to be changed to add a new media system or capability, the IEEE issues a supplement which contains one or more sections, or "clauses" in IEEE-speak. The supplement may consist of one or more entirely new clauses, and may also contain changes to existing clauses in the standard. New supplements to the standard are evaluated by the engineering experts at various IEEE meetings and the supplements must pass a balloting procedure before being voted into the full standard.

New supplements are given a letter designation when they are created. Once the supplement has completed the standardization process, it becomes part of the base standard and is no longer published as a separate supplementary document. On the other hand, you will sometimes see trade literature that refers to Ethernet equipment with the letter of the supplement in which the variety was first developed (e.g., IEEE 802.3u may be used as a reference for Fast Ethernet). Table 1-1 lists several supplements and what they refer to. The dates indicate when formal acceptance of the supplement into the standard occurred. Access to the complete set of supplements is provided in Appendix A, *Resources.*

Table 1-1. IEEE 802.3 Supplements

Supplement	Description
802.3a-1985	10BASE2 thin Ethernet
802.3c-1985	10 Mbps repeater specifications, clause 9
802.3d-1987	FOIRL fiber link
802.3i-1990	10BASE-T twisted-pair

Table 1-1. IEEE 802.3 Supplements (continued)

Supplement	Description
802.3j-1993	10BASE-F fiber optic
802.3u-1995	100BASE-T Fast Ethernet and Auto-Negotiation
802.3x-1997	Full-Duplex standard
802.3z-1998	1000BASE-X Gigabit Ethernet
802.3ab-1999	1000BASE-T Gigabit Ethernet over twisted-pair
802.3ac-1998	Frame size extension to 1522 bytes for VLAN tag
802.3ad-2000	Link aggregation for parallel links

If you've been using Ethernet for a while, you may recall times when a new variety of Ethernet equipment was being sold before the supplement that described the new variety had been entirely completed or voted on. This illustrates a common problem: innovation in the computer field, and especially in computer networking, always outpaces the more deliberate and slow-paced process of developing and publishing standards. Vendors are eager to develop and market new products, and it's up to you, the customer, to make sure that the product will work properly in your network system. One way you can do that is to insist on complete information from the vendor as to what standard the product complies with.

It may not be a bad thing if the product is built to a draft version of a new supplement. Draft versions of the supplements can be substantially complete yet still take months to be voted on by the various IEEE committees. When buying pre-standard equipment built to a draft of the specification, you need to ensure that the draft in question is sufficiently well along in the standards process that no major changes will be made. Otherwise, you could be left out in the cold with network equipment that won't interoperate with newer devices that are built according to the final published standard. One solution to this is to get a written guarantee from the vendor that the equipment you purchase will be upgraded to meet the final published form of the standard. Note that the IEEE forbids vendors to claim or advertise that a product is compliant with an unapproved draft.

Differences in the Standard

When the IEEE adopted the original DIX standard it made a few changes in the specifications. The major reason for the changes made between the DIX and IEEE standards is that the two groups had different goals. The specifications for the original DIX Ethernet standard were developed by the three companies involved and were intended to describe the Ethernet system—and only the Ethernet system. At the time the multi-vendor DIX consortium was developing the first Ethernet standard, there was no open LAN market, nor was there any other multi-vendor

LAN standard in existence. The efforts aimed at creating a worldwide system of open standards had only just begun.

On the other hand, the IEEE was developing standards for integration into the world of international LAN standards. Consequently, the IEEE made several technical changes required for inclusion in the worldwide standardization effort. The IEEE specifications permit backward compatibility with early Ethernet systems built according to the original DIX specifications.* The goal is to standardize network technologies under one umbrella, coordinated with the International Organization for Standardization.

Organization of IEEE Standards

The IEEE standards are organized according to the Open Systems Interconnection (OSI) Reference Model. This model was developed in 1978 by the International Organization for Standardization, whose initials (derived from its French name) are ISO. Headquartered in Geneva, Switzerland, the ISO is responsible for setting open, vendor-neutral standards and specifications for items of technical importance. For example, if you're a photographer you've no doubt noticed the ISO standard speeds for camera film.

The ISO developed the OSI reference model to provide a common organizational scheme for network standardization efforts (with perhaps an additional goal of keeping us all confused with reversible acronyms). What follows is a quick, and necessarily incomplete, introduction to the subject of network models and international standardization efforts.

The Seven Layers of OSI

The OSI reference model is a method of describing how the interlocking sets of networking hardware and software can be organized to work together in the networking world. In effect, the OSI model provides a way to arbitrarily divide the task of networking into separate chunks, which are then subjected to the formal process of standardization.

To do this, the OSI reference model describes seven layers of networking functions, as illustrated in Figure 1-2. The lower layers cover the standards that describe how a LAN system moves bits around. The higher layers deal with more abstract notions, such as the reliability of data transmission and how data is represented to the user. The layers of interest for Ethernet are the lower two layers of the OSI model.

* All Ethernet equipment built since 1985 is based on the IEEE 802.3 standard.

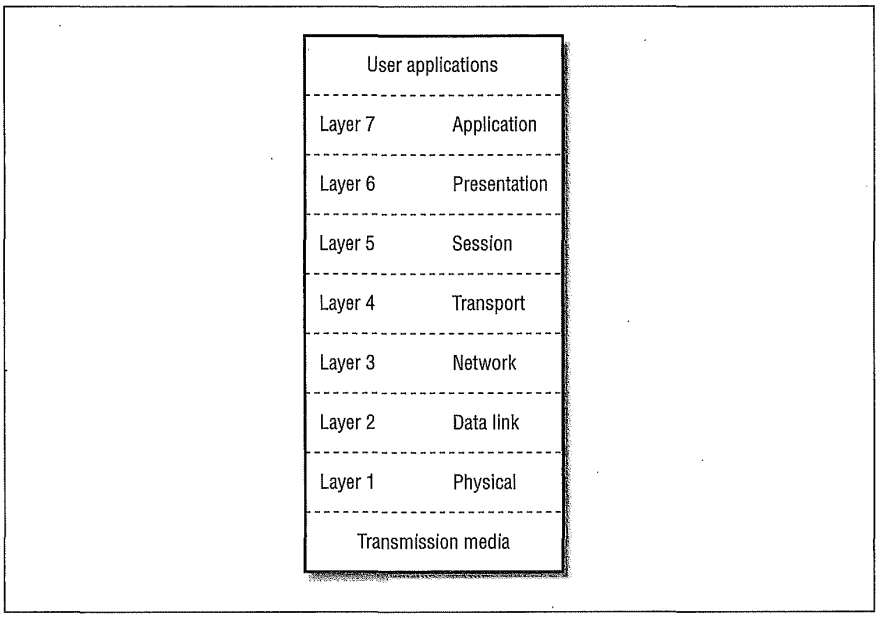


Figure 1-2. The OSI seven layer model

In brief, the OSI reference model includes the following seven layers, starting at the bottom and working our way to the topmost layer:

Physical layer
: Standardizes the electrical, mechanical, and functional control of data circuits that connect to physical media.

Data link layer
: Establishes communication from station to station across a single link. This is the layer that transmits and receives frames, recognizes link addresses, etc. The part of the standard that describes the Ethernet frame format and MAC protocol belongs to this layer.

Network layer
: Establishes communication from station to station across an internetwork, which is composed of a number of data links. This layer provides a level of independence from the lower two layers by establishing higher level functions and procedures for exchanging data between computers across multiple links. Standards at this layer of the model describe portions of the high-level network protocols that are carried over an Ethernet in the data field of the Ethernet frame. Protocols at this layer of the OSI model and above are independent of the Ethernet standard.

Transport layer
: Provides reliable end-to-end error recovery mechanisms and flow control in the higher level networking software.

Session layer
: Provides mechanisms for establishing reliable communications between cooperating applications.

Presentation layer
: Provides mechanisms for dealing with data representation in applications.

Application layer
: Provides mechanisms to support end-user applications such as mail, file transfer, etc.

IEEE Layers Within the OSI Model

The Ethernet standard concerns itself with elements described in Layer 2 and Layer 1, which include the data link layer of the OSI model and below. For that reason, you'll sometimes hear Ethernet referred to as a *link layer standard.*

The Ethernet standards describe a number of entities that all fit within the data link and physical layers of the OSI model. To help organize the details, the IEEE defines extra sublayers that fit into the lower two layers of the OSI model, which simply means that the IEEE standard includes some more finely grained layering than the OSI model.

While at first glance these extra layers might seem to be outside the OSI reference model, the OSI model is not meant to rigidly dictate the structure of network standards. Instead, the OSI model is an organizational and explanatory tool; sublayers can be added to deal with the complexity of a given standard.

Figure 1-3 depicts the lower two layers of the OSI reference model, and shows how the major IEEE-specific sublayers are organized. Within these major sublayers there are even further sublayers defined for additional MAC functions, new physical signaling standards, and so on. At the data link level, there are the Logical Link Control (LLC) and the MAC sublayers, which are the same for all varieties and speeds of Ethernet. The LLC layer is an IEEE mechanism for identifying the data carried in an Ethernet frame. The MAC layer defines the protocol used to arbitrate access to the Ethernet system. Both of these systems are described in detail in Chapter 3, *The Media Access Control Protocol.*

At the physical layer, the IEEE sublayers vary depending on whether 10-, 100-, or 1000 Mbps Ethernet is being standardized. Each of the sublayers is used to help organize the Ethernet specifications around specific functions that must be achieved to make the Ethernet system work.

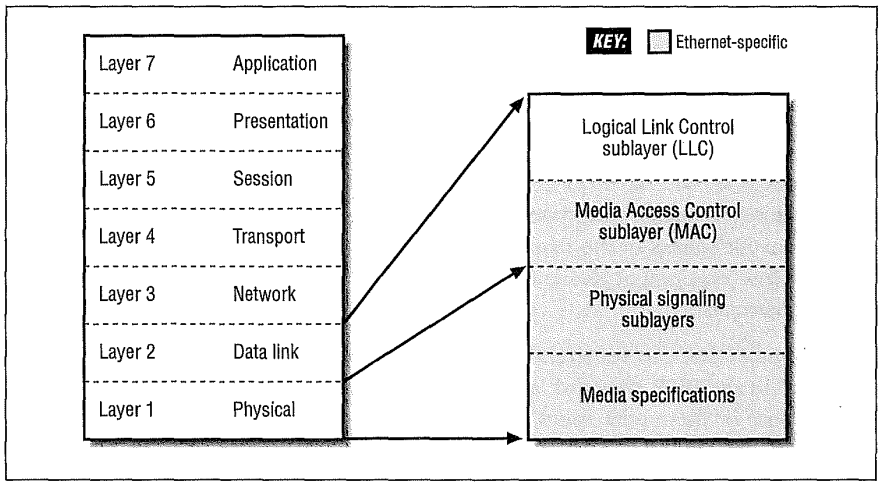


Figure 1-3. The major IEEE sublayers

Understanding these sublayers can also help us understand the scope of the various standards involved. For example, the MAC portion of the IEEE standard is "above" the lower layer physical specifications. As such, it is functionally independent of the various physical layer media specifications and does not change, no matter which physical media variety may be in use.

The IEEE LLC standard is independent of the 802.3 Ethernet LAN standard and will not vary—no matter which LAN system is used. The LLC control fields are intended for use in all LAN systems and not just Ethernet, which is why the LLC sublayer is not formally part of the IEEE 802.3 system specifications.

All of the sublayers below the LLC sublayer are specific to the individual LAN technology in question, which in this case is Ethernet. To help make this clearer, the Ethernet-specific portions of the standard in Figure 1-3 are all shown in gray.

Below the MAC sublayer, we get into the portions of the standard that are organized in the Physical Layer of the OSI reference model. The physical layer standards are different depending on the Ethernet media variety in use and whether or not we're describing the original 10 Mbps Ethernet system, 100 Mbps Fast Ethernet, or 1000 Mbps Gigabit Ethernet system.

Levels of Compliance

In developing a technical standard, the IEEE is careful to include only those items whose behavior must be carefully specified to make the system work. Therefore, all Ethernet interfaces that operate in the original half-duplex mode (described in

Chapter 3) must comply fully with the MAC protocol specifications in the standard to perform the functions identically. Otherwise, the network would not function correctly.

At the same time, the IEEE makes an effort not to constrain the market by standardizing such things as the appearance of an Ethernet interface, or how many connectors it should have on it. The intent is to provide just enough engineering specifications to make the system work reliably, without inhibiting competition and the inventiveness of the marketplace. In general, the IEEE has been quite successful. Most equipment designed for use in an Ethernet system fully complies with the standard.

Vendor innovation can sometimes lead to the development of devices that are not described in the IEEE standard, and that are not included in the half-duplex mode timing specs or the media specs in the standard. Some of these devices may work well for a small network, but might cause problems with signal timing in a larger network operating in half-duplex mode. Further, a network system using equipment not described in the standard or included in the official guidelines cannot be evaluated using the IEEE half-duplex mode configuration guidelines.

The Effect of Standards Compliance

How much you should be concerned about all this is largely up to you and your particular circumstances. Another way of saying this is: "Optimality differs according to context."* It's up to you to decide how important these issues are, given your particular circumstances (or *context*). For one thing, not all innovations are a bad idea.

After all, the thin coaxial and twisted-pair Ethernet media systems started life as vendor innovations that later became carefully specified media systems in the IEEE standard. However, if your goal is maximum predictability and stability for your network given a variety of vendor equipment and traffic loads, then one way to help achieve that goal is by using only equipment that is described in the standard.

One way to decide how important these issues are is to look at the scope and type of network system in question. For an Ethernet that just connects a couple of computers in your house, you may feel that any equipment you can find that helps make this happen at the least cost is a good deal. If the equipment isn't described in the official half-duplex configuration guidelines, you may not care all that much. In this instance, you are building a small network system, and you

* I am indebted to Mike Padlipsky for this useful advice, which was published in his book, *The Elements of Networking Style*, M. A. Padlipsky (Englewood Cliffs, New Jersey: Prentice Hall, 1985), p. 229.

probably don't intend for the network to grow very large. The limited scope of your network makes it easier to decide that you are not all that worried about multi-vendor interoperability, or about your ability to evaluate the network using the IEEE configuration guidelines.

On the other hand, if you are a network manager of a departmental or campus network system, then the people using your network will be depending on the network to get their work done. The expanded scope changes your context quite a bit. Departmental and workgroup nets always seem to be growing, which makes extending networks to accommodate growth a major priority for you. In addition, network stability under all sorts of traffic loads becomes another important issue. In this very different context, the issues of multi-vendor interoperability and compliance with the standard become much more important.

Equipment Included in the Standard

All Ethernet equipment sold is compliant in some way with the standard; otherwise, it wouldn't be able to interoperate with other Ethernet equipment. Therefore, mere compliance with the standard doesn't tell you much. Unfortunately, there's no LAN industry organization that will certify and stamp equipment, "This device is described in the standard and included in the official IEEE configuration guidelines." That's why you need to be wary about believing everything you read in equipment catalogs.

Sometimes vendors may not tell you whether the component they are selling is included in the IEEE system configuration guidelines, and whether it is a piece of standard and interoperable equipment that is widely available from other vendors. Some components that are not included in the official standard or media system configuration guidelines include the 10 Mbps AUI port concentrator, media converters, and special media segments. These components are described in later chapters and Appendix C, *AUI Equipment: Installation and Configuration.*

IEEE Identifiers

The IEEE has assigned shorthand identifiers to the various Ethernet media systems as they have been developed. The three-part identifiers include the speed, the type of signaling used, and information about the physical medium.

In the early media systems, the physical medium part of the identifier was based on the cable distance in meters, rounded to the nearest 100 meters. In the more recent media systems, the IEEE engineers dropped the distance convention and the third part of the identifier simply identifies the media type used (twisted-pair

or fiber optic). In roughly chronological order, the identifiers include the following set:

10BASE5

This identifies the original Ethernet system, based on thick coaxial cable. The identifier means *10* megabits per second transmission speed, *base*band transmission, and the *5* refers to the 500 meter maximum segment length. The word *baseband* simply means that the transmission medium, thick coaxial cable in this instance, is dedicated to carrying one service: Ethernet signals. The 500 meter limit refers to the maximum length a given cable segment may be. Longer networks are built by connecting multiple segments with repeaters or switching hubs.

10BASE2

Also known as the *thin Ethernet* system, this media variety operates at 10 Mbps, in baseband mode, with cable segment lengths that can be a maximum of 185 meters in length. If the segments can be at most 185 meters long, then why does the identifier say "2," thus implying a maximum of 200 meters? The answer is that the identifier is merely a bit of shorthand and not intended to be an official specification. The IEEE committee found it convenient to round things up to 2, to keep the identifier short and easier to pronounce. This less expensive version of coax Ethernet was nicknamed "Cheapernet."

FOIRL

This stands for *Fiber Optic Inter-Repeater Link*. The original DIX Ethernet standard mentioned a point-to-point link segment that could be used between repeaters, but did not provide any media specifications. Later, the IEEE committee developed the FOIRL standard, and published it in 1989. FOIRL segments were originally designed to link remote Ethernet segments together. Fiber optic media's immunity to lightning strikes and electrical interference, as well as its ability to carry signals for long distances, makes it an ideal system for transmitting signals between buildings.

The specifications in the original FOIRL segment only provide for linking two repeaters together, one at each end of the link. While waiting for a larger set of fiber optic specifications to appear, vendors extended the set of devices that are connected via fiber, allowing an FOIRL segment to be attached to a station as well. These changes were taken up and added to the newer fiber optic link specifications found in the 10BASE-F standard (described later in this section).

10BROAD36

This system was designed to send 10 Mbps signals over a broadband cable system. Broadband cable systems support multiple services on a single cable by dividing the bandwidth of the cable into separate frequencies, each

assigned to a given service. Cable television is an example of a broadband cable system, designed to deliver multiple television channels over a single cable. 10BROAD36 systems are intended to cover a large area; the 36 refers to the 3,600 meter distance allowed between any two stations on the system. These days, the vast majority of sites use fiber optic media for covering large distances, and broadband Ethernet equipment is not widely available.

1BASE5

This standard describes a 1 Mbps system based on twisted-pair wiring, which did not prove to be a very popular system. 1BASE5 was superseded in the marketplace by 10BASE-T, which provided all the advantages of twisted-pair wiring as well as the higher 10 Mbps speed.

10BASE-T

The "T" stands for "twisted," as in twisted-pair wires. This variety of the Ethernet system operates at 10 Mbps, in baseband mode, over two pairs of Category 3 (or better) twisted-pair wires. The category system for classifying cable quality is described in Chapter 14, *Structured Cabling*. The hyphen was added to the "10BASE-T" identifier to help ensure the correct pronunciation of "ten base tee." It was felt that without the hyphen people might mistakenly call it "10 basset," which is too close to the dog, "basset hound." Use of the hyphen is found in this and all newer media identifiers.

10BASE-F

The "F" stands for *fiber*, as in *fiber optic media*. This is the most recent 10 Mbps fiber optic Ethernet standard, adopted as an official part of the IEEE 802.3 standard in November 1993. The 10BASE-F standard defines three sets of specifications:

10BASE-FB

This is for active fiber hubs based on synchronous repeaters for extending a backbone system.

10BASE-FP

This is for passive hub equipment intended to link workstations with a fiber optic hub.

10BASE-FL

This includes a set of fiber optic link segment specifications that updates and extends the older FOIRL standard.

Two of these specifications have not been widely deployed. Equipment based on 10BASE-FB is scarce, and equipment based on 10BASE-FP is non-existent. The vast majority of Ethernet vendors sell 10BASE-FL fiber link equipment.

100 Mbps Media Systems

100BASE-T

This is the IEEE shorthand identifier for the entire 100 Mbps system, including all twisted-pair and fiber optic Fast Ethernet media systems.

100BASE-X

This is the IEEE shorthand identifier for the 100BASE-TX and 100BASE-FX media systems. These two systems are both based on the same 4B/5B block encoding system, adapted from a 100 Mbps networking standard called Fiber Distributed Data Interface (FDDI). FDDI was originally developed and standardized by ANSI.

100BASE-TX

This variety of the Fast Ethernet system operates at 100 Mbps, in baseband mode, over two pairs of high-quality, Category 5 twisted-pair cable. The TX identifier indicates that this is the twisted-pair version of the 100BASE-X media systems. This is the most widely used variety of Fast Ethernet.

100BASE-FX

This variety of the Fast Ethernet system operates at 100 Mbps, in baseband mode, over multi-mode fiber optic cable.

100BASE-T4

This variety of the Fast Ethernet system operates at 100 Mbps, in baseband mode, over four pairs of Category 3 or better twisted-pair cable. This variety has not been widely deployed, and 100BASE-T4 equipment is scarce.

100BASE-T2

This variety of the Fast Ethernet system operates at 100 Mbps, in baseband mode, on two pairs of Category 3 or better twisted-pair cable. This variety was never developed by any vendor, and equipment based on the T2 standard is non-existent.

1000 Mbps Media Systems

1000BASE-X

This is the IEEE shorthand identifier for the Gigabit Ethernet media systems based on the 8B/10B block encoding scheme adapted from the Fibre Channel networking standard. Fibre Channel is a high speed networking system developed and standardized by ANSI.

The 1000BASE-X media systems include 1000BASE-SX, 1000BASE-LX, and 1000BASE-CX.

1000BASE-SX
: The "S" stands for "short," as in short wavelength. The "X" indicates that this media segment is one of three based on the same block encoding scheme. This is the short wavelength fiber optic media segment for Gigabit Ethernet.

1000BASE-LX
: This is the long wavelength fiber optic media segment for Gigabit Ethernet.

1000BASE-CX
: This is a short copper cable media segment for Gigabit Ethernet, based on the original Fibre Channel standard.

1000BASE-T
: This is the IEEE shorthand identifier for 1000 Mbps Gigabit Ethernet over Category 5 or better twisted-pair cable. This system is based on a different signal encoding scheme required to transmit gigabit signals over twisted-pair cabling.

Reinventing Ethernet

No matter how well designed a LAN system is, it won't help you much if you can only use it with a single vendor's equipment. A LAN has to be able to work with the widest variety of equipment possible to provide you with the greatest flexibility. For maximum utility, your LAN must be vendor-neutral: that is, capable of interworking with all types of computers without being vendor-specific. This was not the way things worked in the 1970s when computers were expensive and networking technology was exotic and proprietary.

Metcalfe understood that a revolution in computer communications required a networking technology that everyone could use. In 1979 he set out to make Ethernet an open standard, and convinced Xerox to join a multi-vendor consortium for the purposes of standardizing an Ethernet system that any company could use. The era of open computer communications based on Ethernet technology formally began in 1980 when the Digital Equipment Corporation (DEC), Intel, and Xerox consortium announced the DIX standard for 10 Mbps Ethernet.

This DIX standard made the technology available to anyone who wanted to use it, producing an open system. As part of this effort, Xerox agreed to license its patented technology for a low fee to anyone who wanted it. In 1982 Xerox also gave up its trademark on the Ethernet name. As a result, the Ethernet standard became the world's first open, multi-vendor LAN standard. The idea of sharing proprietary computer technology in order to arrive at a common standard to benefit everyone was a radical notion for the computer industry in the late 1970s. It's a tribute to

Bob Metcalfe's vision that he realized the importance of making Ethernet an open standard. As Metcalfe put it:

> The invention of Ethernet as an open, non-proprietary, industry-standard local network was perhaps even more significant than the invention of Ethernet technology itself.*

In 1979 Metcalfe started a company to help commercialize Ethernet. He believed that computers from multiple vendors ought to be able to communicate compatibly over a common networking technology, making them more useful and, in turn, opening up a vast new set of capabilities for the users. *Com*puter *com*munication *com*patibility was the goal, which led Metcalfe to name his new company 3Com.

Reinventing Ethernet for Twisted-Pair Media

Ethernet prospered during the 1980s, but as the number of computers being networked continued to grow, the problems inherent in the original coaxial cable media system became more acute. Installing a thick coax cable in a building was a difficult task, and connecting the computers to the cable was also a challenge. A thin coax cable system was introduced in the mid-1980s that made it easier to build a media system and connect computers to it, but it was still difficult to manage Ethernet systems based on coaxial cable. Coaxial Ethernet systems are typically based on a bus topology, in which every computer sends Ethernet signals over a single bus cable; a failure of the cable brings the entire network system to a halt, and troubleshooting a cable problem can take a long time.

The invention of *twisted-pair* Ethernet in the late 1980s by a company called SynOptics Communications made it possible to build Ethernet systems based on the much more reliable star-wired cabling topology, in which the computers are networked to a central hub. These systems are much easier to install and manage, and troubleshooting is much easier and quicker as well. The use of twisted-pair cabling was a major change, or reinvention, of Ethernet. Twisted-pair Ethernet led to a vast expansion in the use of Ethernet; the Ethernet market took off and has never looked back.

In the early 1990s, a structured cabling system standard for twisted-pair cabling systems was developed that made it possible to provide building-wide twisted-pair systems based on high-reliability cabling practices adopted from the telephone industry. Ethernet based on twisted-pair media installed according to the structured cabling standard has become the most widely used network technology.

---

* Shotwell, *The Ethernet Sourcebook*, p. xi.

These Ethernet systems are reliable, easy to install and manage, and support rapid troubleshooting for problem resolution.

Reinventing Ethernet for 100 Mbps

The original Ethernet standard of 1980 described a system that operated at 10 Mbps. This was quite fast for the time, and Ethernet interfaces in the early 1980s were expensive due to the buffer memory and high-speed components required to support such rapid speeds. Throughout the 1980s, Ethernet was considerably faster than the computers connected to it, which provided a good match between the network and the computers it supported. However, as computer technology continued to evolve, ordinary computers were fast enough to provide a major traffic load to a 10 Mbps Ethernet channel by the early 1990s.

Much to the surprise of those who thought Ethernet was limited to 10 Mbps, Ethernet was reinvented to increase its speed by a factor of ten. The new standard created the 100 Mbps Fast Ethernet system, which was formally adopted in 1995. Fast Ethernet is based on twisted-pair and fiber optic media systems, and provides high-speed network channels for use in backbone systems, as well as connections to fast server computers and to desktop computers.

With the invention of Fast Ethernet, multi-speed twisted-pair Ethernet interfaces can be built which operate at either 10 or 100 Mbps. These interfaces are able, through an Auto-Negotiation protocol, to automatically set their speed in interaction with Ethernet repeater hubs and switching hubs. This makes the migration from 10 Mbps to 100 Mbps Ethernet systems easy to accomplish.

Reinventing Ethernet for 1000 Mbps

In 1998, Ethernet was reinvented yet again, this time to increase its speed by another factor of ten. The new Gigabit Ethernet standard describes a system that operates at the speed of 1 billion bits per second over fiber optic and twisted-pair media. The invention of Gigabit Ethernet makes it possible to provide very fast backbone networks as well as connections to high-performance servers.

The twisted-pair standard for Gigabit Ethernet makes it possible to provide very high-speed connections to the desktop when needed. Multi-speed twisted-pair Ethernet interfaces can now be built which operate at 10-, 100-, or 1000 Mbps, using the Auto-Negotiation protocol (see Chapter 5, *Auto-Negotiation*) to automatically configure their speed.

Reinventing Ethernet for New Capabilities

Ethernet innovations include new speeds and new media systems. They also include new Ethernet capabilities. For example, the full-duplex Ethernet standard makes it possible for two devices connected to a full-duplex Ethernet media system to simultaneously send and receive data. A port on a Fast Ethernet switching hub can simultaneously send and receive data at 100 Mbps with a server when using full-duplex mode, resulting in a total link bandwidth of 200 Mbps. The Auto-Negotiation standard provides the ability for switching hub ports and computers linked to those ports to discover whether or not they both support full-duplex mode, and if they do, to automatically select that mode of operation.

As you can see, the Ethernet system has been reinvented again and again to provide more flexible and reliable cabling, to accommodate the rapid increase in network traffic with higher speeds, and to provide more capabilities for today's more complex network systems. The remarkable success of Ethernet in the marketplace has been based on the equally remarkable ability of the system to adapt and change to meet the rapidly evolving needs of the computer industry.

Multi-Gigabit Ethernet

In March 1999, the IEEE 802.3 standards group held a "Call for Interest" meeting on the topic of Ethernet speeds beyond the current 1 Gbps standard. A number of presentations were made on the general topic, after which the group voted to create a High Speed Study Group. At the time of this writing, the High Speed Study Group is meeting on a regular basis to review presentations on a variety of technical issues. It is expected that this work will lead to the development of a new higher speed Ethernet standard, probably operating at 10 Gbps, within the next few years.

In this chapter:

- *Four Basic Elements of Ethernet*
- *Ethernet Hardware*
- *Network Protocols and Ethernet*

2

The Ethernet System

An Ethernet Local Area Network (LAN) is made up of hardware and software working together to deliver digital data between computers. To accomplish this task, four basic elements are combined to make an Ethernet system. This chapter provides a tutorial describing these elements, since a familiarity with these basic elements provides a good background for working with Ethernet. We will also take a look at some network media and simple topologies. Finally, we will see how the Ethernet system is used by high-level network protocols to send data between computers.

This chapter describes the original half-duplex mode of operation. *Half-duplex* simply means that only one computer can send data over the Ethernet channel at any given time. In half-duplex mode, multiple computers share a single Ethernet channel by using the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) media access control (MAC) protocol. Until the introduction of switching hubs, the half-duplex system was the typical mode of operation for the vast majority of Ethernet LANs—tens of millions of Ethernet connections have been installed based on this system.

However, these days many computers are connected directly to their own port on an Ethernet switching hub and do not share the Ethernet channel with other systems. This type of connection is described in Chapter 18, *Ethernet Switching Hubs.* Many computers and switching hub connections now use full-duplex mode, in which the CSMA/CD protocol is shut off and the two devices on the link can send data whenever they like. The full-duplex mode of operation is described in Chapter 4, *Full-Duplex Ethernet.*

Four Basic Elements of Ethernet

The Ethernet system includes four building blocks that, when combined, make a working Ethernet:

- The *frame*, which is a standardized set of bits used to carry data over the system.
- The *media access control protocol*, which consists of a set of rules embedded in each Ethernet interface that allow multiple computers to access the shared Ethernet channel in a fair manner.
- The *signaling components*, which consists of standardized electronic devices that send and receive signals over an Ethernet channel.
- The *physical medium*, which consists of the cables and other hardware used to carry the digital Ethernet signals between computers attached to the network.

The Ethernet Frame

The heart of the Ethernet system is the frame. The network hardware—which is comprised of the Ethernet interfaces, media cables, etc.—exists simply to move Ethernet frames between computers, or stations.* The bits in the Ethernet frame are formed up in specified fields. Figure 2-1 shows the basic frame fields. These fields are described in more detail in Chapter 3, *The Media Access Control Protocol.*

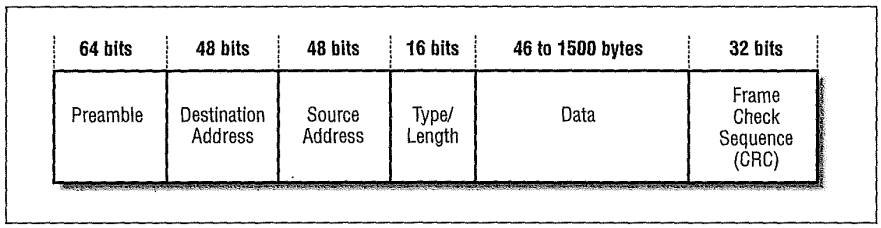


Figure 2-1. An Ethernet frame

Figure 2-1 shows the basic Ethernet frame, which begins with a set of 64 bits called the *preamble.* The preamble gives all of the hardware and electronics in a 10 Mbps Ethernet system some signal start-up time to recognize that a frame is being transmitted, alerting it to start receiving the data. This is what a 10 Mbps network uses to clear its throat, so to speak. Newer Ethernet systems running at 100 and 1000 Mbps use constant signaling, which avoids the need for a preamble.

* A computer connected to the network may be a standard desktop workstation, a printer, or anything else with an Ethernet interface in it. For that reason, the Ethernet standard uses the more general term "station" to describe the networked device, and so will we.

However, the preamble is still transmitted in these systems to avoid making any changes in the structure of the frame.

Following the preamble are the destination and source addresses. Assignment of addresses is controlled by the IEEE Standards Association (IEEE-SA), which administers a portion of the address field. When assigning blocks of addresses for use by network vendors, the IEEE-SA provides a 24-bit Organizationally Unique Identifier (OUI).* The OUI is a unique 24-bit identifier assigned to each organization that builds network interfaces. This allows a vendor of Ethernet equipment to provide a unique address for each interface they build. Providing unique addresses during manufacturing avoids the problem of two or more Ethernet interfaces in a network having the same address. This also eliminates any need to locally administer and manage Ethernet addresses.

A manufacturer of Ethernet interfaces creates a unique 48-bit Ethernet address for each interface by using their assigned OUI for the first 24 bits of the address. The vendor then assigns the next 24 bits, being careful to ensure that each address is unique. The resulting 48-bit address is often called the hardware, or physical, address to make the point that the address has been assigned to the Ethernet interface. It is also called a Media Access Control (MAC) address, since the Ethernet media access control system includes the frame and its addressing.

Following the addresses in an Ethernet frame is a 16-bit type or length field. Most often this field is used to identify what type of high-level network protocol is being carried in the data field, e.g., TCP/IP. This field may also be used to carry length information, as described in Chapter 3.

After the type field can come anywhere from 46 bytes to 1500 bytes of data. The data field *must* be at least 46 bytes long. This length assures that the frame signals stay on the network long enough for every Ethernet station on the network system to hear the frame within the correct time limits. Every station must hear the frame within the maximum round-trip signal propagation time of an Ethernet system, as described later in this chapter. If the high-level protocol data carried in the data field is shorter than 46 bytes, then padding data is used to fill out the data field.

Finally, at the end of the frame there's a 32-bit Frame Check Sequence (FCS) field. The FCS contains a Cyclic Redundancy Checksum (CRC) which provides a check of the integrity of the data in the entire frame. The CRC is a unique number that is generated by applying a polynomial to the pattern of bits that make up the frame. The same polynomial is used to generate another checksum at the receiving station. The receiving station checksum is then compared to the checksum generated

* The IEEE-SA web page for OUI assignment is listed in Appendix A, *Resources.*

at the sending station. This allows the receiving Ethernet interface to verify that the bits in the frame survived their trip over the network system intact.

That's basically all there is to an Ethernet frame. Now that you know what an Ethernet frame looks like, you need to know how the frames are transmitted. This is where the set of rules used to govern when a station gets to transmit a frame comes into play. We'll take a look at those rules next.

The Media Access Control Protocol

The half-duplex mode of operation described in the original Ethernet standard uses the MAC protocol, which is a set of rules used to arbitrate access to the shared channel among a set of stations connected to that channel. The way the access protocol works is fairly simple: each Ethernet-equipped computer operates independently of all other stations on the network; there is no central controller. All stations attached to an Ethernet operating in half-duplex mode are connected to a shared signaling channel, also known as a signal bus.

Ethernet uses a *broadcast delivery* mechanism, in which each frame that is transmitted is heard by every station. While this may seem inefficient, the advantage is that putting the address-matching intelligence in the interface of each station allows the physical medium to be kept as simple as possible. On an Ethernet LAN, all that the physical signaling and media system has to do is see that the bits are accurately transmitted to every station; the Ethernet interface in the station does the rest of the work.

Ethernet signals are transmitted from the interface and sent over the shared signal channel to every attached station. To send data, a station first listens to the channel, and if the channel is idle the station transmits its data in the form of an Ethernet frame or packet.*

As each Ethernet frame is sent over the shared signal channel, or medium, all Ethernet interfaces connected to the channel read in the bits of the signal and look at the second field of the frame shown in Figure 2-1, which contains the destination address. The interfaces compare the destination address of the frame with their own 48-bit unicast address or a multicast address they have been enabled to recognize. The Ethernet interface whose address matches the destination address in the frame will continue to read the entire frame and deliver it to the networking software running on that computer. All other network interfaces will stop reading the frame when they discover that the destination address does not match their own unicast address or an enabled multicast address.

* The precise term as defined in the Ethernet standard is "frame," but the term "packet" is sometimes used as well.

Multicast and Broadcast Addresses

A *multicast* address allows a single Ethernet frame to be received by a group of stations. An application providing streaming audio and video services, for example, can set a station's Ethernet interface to listen for specific multicast addresses. This makes it possible for a set of stations to be assigned to a multicast group, which has been given a specific multicast address. A single stream of audio packets sent to the multicast address assigned to that group will be received by all stations in that group.

There is also the special case of the multicast address known as the *broadcast* address, which is the 48-bit address of all ones. All Ethernet interfaces that see a frame with this destination address will read the frame in and deliver it to the networking software on the computer.

After each frame transmission, all stations on the network with traffic to send must contend equally for the next frame transmission opportunity. This ensures that access to the network channel is fair and that no single station can lock out the others. Fair access of the shared channel is made possible through the use of the MAC system embedded in the Ethernet interface located in each station. The media access control mechanism for Ethernet uses the CSMA/CD protocol.

The CSMA/CD protocol

The CSMA/CD protocol functions somewhat like a dinner party in a dark room where the participants can only hear one another. Everyone around the table must listen for a period of quiet before speaking (*Carrier Sense*). Once a space occurs everyone has an equal chance to say something (*Multiple Access*). If two people start talking at the same instant they detect that fact, and quit speaking (*Collision Detection*).

To translate this into Ethernet terms, the Carrier Sense portion of the protocol means that before transmitting, each interface must wait until there is no signal on the channel. If there is no signal, it can begin transmitting. If another interface is transmitting, there will be a signal on the channel; this condition is called *carrier.** All other interfaces must wait until carrier ceases and the signal channel is idle before trying to transmit; this process is called *deferral.*

* Historically, a carrier signal is defined as a continuous constant-frequency signal, such as the one used to carry the modulated signal in an AM or FM radio system. There is no such continuous carrier signal in Ethernet; instead, "carrier" in Ethernet simply means the presence of traffic on the network.

With Multiple Access, all Ethernet interfaces have the same priority when it comes to sending frames on the network, and all interfaces can attempt to access the channel at any time.

The next portion of the access protocol is called Collision Detect. Given that every Ethernet interface has equal opportunity to access the Ethernet, it's possible for multiple interfaces to sense that the network is idle and start transmitting their frames simultaneously. When this happens, the Ethernet signaling devices connected to the shared channel sense the *collision* of signals, which tells the Ethernet interfaces to stop transmitting. Each of the interfaces will then choose a random retransmission time and resend their frames in a process called *backoff.*

The CSMA/CD protocol is designed to provide fair access to the shared channel so that all stations get a chance to use the network and no station gets locked out due to some other station hogging the channel. After every packet transmission, all stations use the CSMA/CD protocol to determine which station gets to use the Ethernet channel next.

Collisions

If more than one station happens to transmit on the Ethernet channel at the same moment, then the signals are said to *collide.* The stations are notified of this event and reschedule their transmission using a random time interval chosen by a specially designed backoff algorithm. Choosing random times to retransmit helps the stations to avoid colliding again on the next transmission.

It's unfortunate that the original Ethernet design used the word *collision* for this aspect of the Ethernet media access control mechanism. If it had been called something else, such as Distributed Bus Arbitration (DBA) events, then no one would worry about the occurrence of DBAs on an Ethernet. To most ears the word "collision" sounds like something bad has happened, leading many people to incorrectly conclude that collisions are an indication of network failure and that lots of collisions must mean the network is broken.

Instead, the truth of the matter is that collisions are absolutely normal events on an Ethernet and are simply an indication that the CSMA/CD protocol is functioning as designed. As more computers are added to a given Ethernet, there will be more traffic, resulting in more collisions as part of the normal operation of an Ethernet. Collisions resolve quickly. For example, the design of the CSMA/CD protocol ensures that the majority of collisions on a 10 Mbps Ethernet will be resolved in microseconds, or millionths of a second. Nor does a normal collision result in lost data. In the event of a collision, the Ethernet interface backs off (waits) for some number of microseconds, and then automatically retransmits the frame.

Networks with very heavy traffic loads may experience multiple collisions for each frame transmission attempt. This is also expected behavior. Repeated collisions for a given packet transmission attempt indicate a very busy network. If repeated collisions occur, the stations involved will expand the set of potential backoff times in order to retransmit the data. The expanding backoff process, formally known as *truncated binary exponential backoff*, is a clever feature of the Ethernet MAC protocol that provides an automatic method for stations to adjust to changing traffic conditions on the network. Only after 16 consecutive collisions for a given transmission attempt will the interface finally discard the Ethernet frame. This can happen only if the Ethernet channel is overloaded for a fairly long period of time or if it is broken.

So far, we've seen what an Ethernet frame looks like and how the CSMA/CD protocol is used to ensure fair access for multiple stations sending their frames over the shared Ethernet channel. The frame and the CSMA/CD protocol are the same for all varieties of Ethernet. Whether the Ethernet signals are carried over coaxial, twisted-pair, or fiber optic cable, the same frame is used to carry the data and the same CSMA/CD protocol is used to provide the half-duplex shared channel mode of operation. In full-duplex mode, the same frame format is used, but the CSMA/CD protocol is shut off, as described in Chapter 4.

Ethernet Hardware

The next two building blocks of an Ethernet system include the hardware components used in the system. There are two basic groups of hardware components: the signaling components, used to send and receive signals over the physical medium; and the media components, used to build the physical medium that carries the Ethernet signals. Not surprisingly, these hardware components differ depending on the speed of the Ethernet system and the type of cabling used. To show the hardware building blocks, we'll look at an example based on the widely used 10 Mbps twisted-pair Ethernet media system, called 10BASE-T.

Signaling Components

The signaling components for a twisted-pair system include the Ethernet interface located in the computer, as well as a transceiver and its cable. An Ethernet may consist of a pair of stations linked with a single twisted-pair segment, or multiple stations connected to twisted-pair segments that are linked together with an Ethernet repeater. A repeater is a device used to repeat network signals onto multiple segments. Connecting cable segments with a repeater makes it possible for the segments to all work together as a single shared Ethernet channel.

Figure 2-2 shows two computers (*stations*) connected to a 10BASE-T media system. Both computers have an Ethernet interface card installed, which makes the Ethernet system operate. The interface contains the electronics needed to form up and send Ethernet frames, as well as to receive frames and extract the data from them. The Ethernet interface comes in two basic types. The first is a board that plugs into a computer's bus slot, and the other relies on chips that allow Ethernet interfaces to be built into the computer's main logic board. In the second form, all you'll see of the interface is an Ethernet connector mounted on the back of the computer.

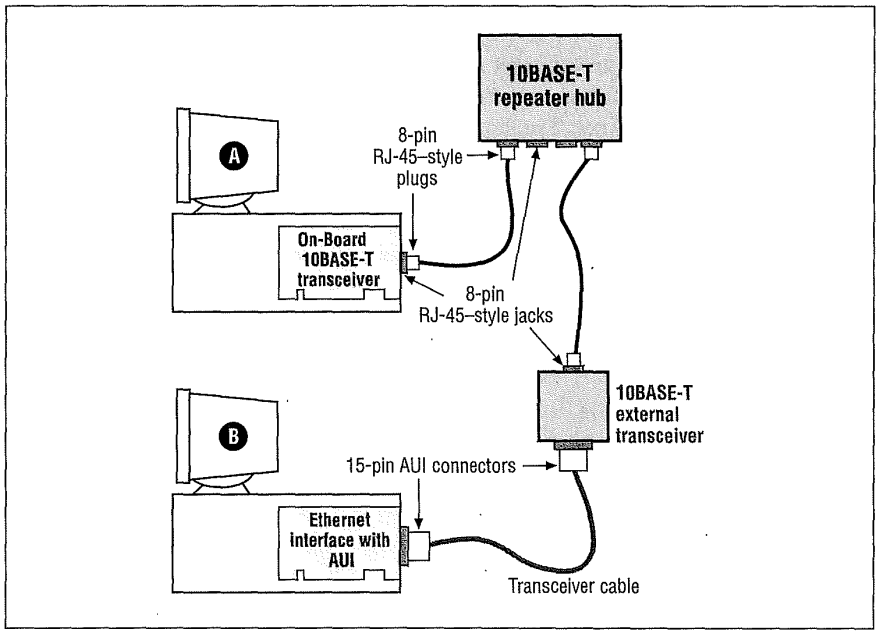


Figure 2-2. A sample 10BASE-T Ethernet connection

The Ethernet interface connects to the media system using a transceiver, which can be built into the interface or provided as an external device. Of the two stations shown in Figure 2-2, one is provided with a built-in transceiver and one uses an external transceiver. The word "transceiver" is a combination of *trans*mitter and re*ceiver*. A transceiver contains the electronics needed to take signals from the station interface and transmit them to the twisted-pair cable segment, and to receive signals from the cable segment and send them to the interface.

The Ethernet interface in Station A is connected directly to the twisted-pair cable segment since it is equipped with an internal 10BASE-T transceiver. The twisted-

pair cable uses an 8-pin connector that is also known as an RJ-45 plug. The Ethernet interface in Station B is connected to an outboard transceiver, which is a small box that contains the transceiver electronics. The outboard transceiver connects to the twisted-pair segment using an 8-pin connector. The outboard transceiver connects to the Ethernet interface in the station with a transceiver cable. The transceiver cable in the 10 Mbps Ethernet system uses a 15-pin connector which is called the Attachment Unit Interface (AUI).

The final signaling component shown in Figure 2-2 is an Ethernet repeater hub which links multiple twisted-pair segments. This device is called a hub because it sits at the center, or hub, of a set of cable segments. The repeater connects to the cable segments using the same built-in 10BASE-T transceivers used by an ordinary station interface. The repeater operates by moving Ethernet signals from one segment to another one bit at a time; it does not operate at the level of Ethernet frames, but simply repeats the signals it sees on each segment. Repeaters make it possible to build larger Ethernet systems composed of multiple cable segments by making the segments function together as a single channel.

Media Components

The cables and other components used to build the signal-carrying portion of the shared Ethernet channel are called the *physical* media. The physical cabling components vary depending on which kind of media system is in use. For instance, a twisted-pair cabling system uses different components than a fiber optic cabling system. Just to make things more interesting, a given Ethernet system may include several different kinds of media systems all connected together with repeaters to make a single network channel.

By using repeaters, an Ethernet system of multiple segments can grow as a *branching tree*. This means that each media segment is an individual branch of the complete signal system. The 10 Mbps system allows multiple repeaters in the path between stations, making it possible to build repeated networks with multiple branches. Only one or two repeaters can be used in the path of higher speed Ethernet systems, limiting the size of the resulting network. A typical network design actually ends up looking less like a tree and more like a complex set of network segments that may be strung along hallways or throughout wiring closets in your building. The resulting system of connected segments may grow in any direction and does not have a specific root segment. However, it is essential not to connect Ethernet segments in a loop, as each frame would circulate endlessly until the system was saturated with traffic.

Round-Trip Timing

In order for the MAC system to work properly, all Ethernet interfaces must be capable of responding to one another's signals within a strictly controlled amount of time. The signal timing for Ethernet is based on the amount of time it takes for a signal to get from one end of the complete media system to the other and back. This is known as the *round-trip time.* The maximum round-trip time of signals on an Ethernet system operating in half-duplex mode is strictly limited. Limiting the round-trip time ensures that every interface can hear all network signals within the specified amount of time provided for in the Ethernet MAC system.

The longer a given network segment is, the more time it takes for a signal to travel over it. The intent of the configuration guidelines in the standard is to make sure that the round-trip timing restrictions are met, no matter what combination of media segments are used in your network. The configuration guidelines provide specifications for the maximum length of segments and rules for combining various kinds of cabling segments with repeaters. These specifications and rules ensure that the correct signal timing is maintained for the entire LAN. The specifications for individual media segment lengths and the rules for combining segments must be carefully followed. If these specifications and rules are violated, the computers may not hear one another's signals within the required time limit, and could end up interfering with one another.

That's why the correct operation of an Ethernet LAN depends upon media segments that are built according to the rules published for each media type. More complex LANs built with multiple media types must be designed according to the multi-segment configuration guidelines provided in the Ethernet standard. These rules include limits on the total length of segments and the total number of segments and repeaters that may be in a given system. This is to ensure that the correct round-trip timing is maintained.

Ethernet Hubs

Ethernet was designed to be easily expandable to meet the networking needs of a given site. As we've just seen, the total set of segments and repeaters in the Ethernet LAN must meet round-trip timing specifications. To help extend half-duplex Ethernet systems, networking vendors sell repeater hubs that are equipped with multiple Ethernet ports. Each port of a repeater hub links individual Ethernet media segments together to create a larger network that operates as a single Ethernet LAN.

There is another kind of hub called a switching hub. Switching hubs use the 48-bit Ethernet destination addresses to make a frame forwarding decision from one port of the switch to another. As shown in Figure 2-3, each port of a switching hub

provides a connection to an Ethernet media system that functions as an entirely separate Ethernet LAN.

In a repeater hub the individual ports combine segments together to create a single LAN channel. However, a switching hub makes it possible to divide a set of Ethernet media systems into multiple separate LANs. The separate LANs are linked together by way of the switching electronics in the hub. The round-trip timing rules for each LAN stop at the switching hub port, allowing you to link a large number of individual Ethernet LANs together.

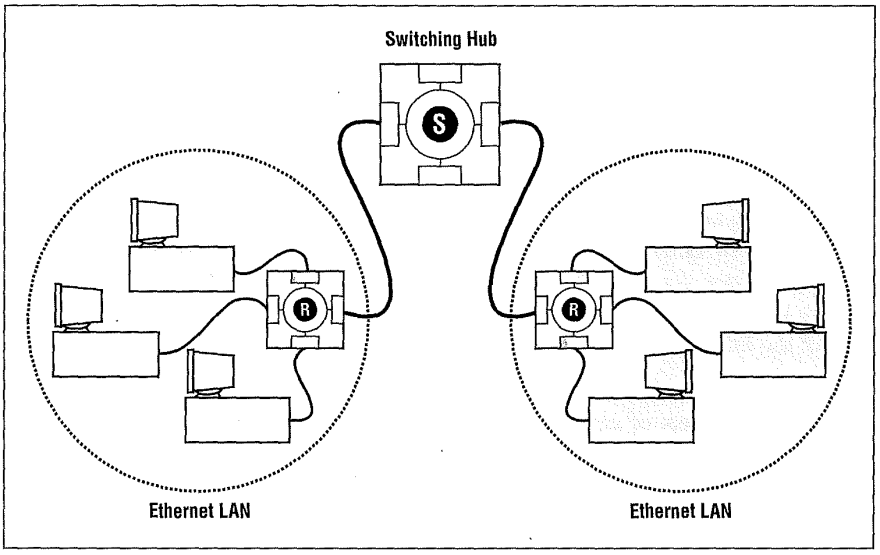


Figure 2-3. Switching hub creates separate Ethernet LANs

A given Ethernet LAN may consist of a repeater hub linking several media segments together. Whole Ethernet LANs can themselves be linked together to form extended network systems using switching hubs. Larger networks based on repeater hubs can be segmented into smaller LANs with switching hubs in a process called *network segmentation.* In this instance, the switching hub is used to segment a single LAN composed of network segments linked by repeater hubs into multiple LANs, to improve bandwidth and reliability.

Network segmentation can be extended all the way to connecting individual stations to single ports on the switching hub, in a process called *micro-segmentation.* As switching hub costs have dropped and computer performance has increased, more and more stations are being connected directly to their own port on the switching hub. That way, the station does not have to share the Ethernet channel bandwidth with another computer. Instead, it has its own dedicated Ethernet link

to the switching hub. Switching hub operation and network segmentation are detailed in Chapter 18.

Network Protocols and Ethernet

Now that we've seen how frames are sent over Ethernet systems, let's look at the data being carried by the frame. Data that is being sent between computers is carried in the data field of the Ethernet frame and structured as high-level network protocols. The high-level network protocol information carried inside the data field of each Ethernet frame is what actually establishes communications between applications running on computers attached to the network. The most widely used system of high-level network protocols is called the Transmission Control Protocol/Internet Protocol (TCP/IP) suite.

The important thing to understand is that the high-level protocols are *independent* of the Ethernet system. There are several network protocols in use today, any of which may send data between computers in the data field of an Ethernet frame. In essence, an Ethernet LAN with its hardware and Ethernet frame is simply a trucking service for data being sent by applications. The Ethernet LAN itself doesn't know or care about the high-level protocol data being carried in the data field of the Ethernet frame.

Since the Ethernet system is unaffected by the contents of the data field in the frame, different sets of computers running different high-level network protocols can share the same Ethernet. For example, you can have a single Ethernet that supports four computers, two of which communicate using TCP/IP, and two that use some other system of high-level protocols. All four computers can send Ethernet frames over the same Ethernet system without any problem.

The details of how network protocols function are an entirely separate subject from how the Ethernet system works and are outside the scope of this book. However, Ethernets are installed to make it possible for applications to communicate between computers using high-level network protocols to facilitate the communication. Let's take a quick look at one example of high-level network protocols to see how the Ethernet system and network protocols work together.

Design of Network Protocols

Network protocols are easy to understand since we all use some form of protocol in daily life. For instance, there's a certain protocol to writing a letter. We can compare the act of composing and delivering a letter to what a network protocol does to see how each works. The letter has a well-known form that has been "standardized" through custom. The letter includes a basic message with a greeting to the recipient and the name of the sender. After you're through writing the

letter, you stuff it into an envelope, write the name and address of both the recipient and sender on the envelope, and give it to a delivery system, such as the post office, which handles the details of getting the message to the recipient's address.

A network protocol acts much like the letter protocol described above. To carry data between applications, the network software on your computer creates and sends a network protocol packet with its own private data field that corresponds to the message of the letter. The sender's and recipient's names (or protocol addresses) are added to complete the packet. After the network software has created the packet, the entire network protocol packet is stuffed into the data field of an Ethernet frame. Next, the 48-bit destination and source addresses are provided, and the frame is handed to the Ethernet interface and the Ethernet signal and cabling system for delivery to the right computer.

Figure 2-4 shows network protocol data traveling from Station A to Station B. The data is depicted as a letter that is placed in an envelope (i.e., a high-level protocol packet) that has network protocol addresses on it. This letter is stuffed into an Ethernet frame, shown here as a mailbag. The analogy is not exact, in that each Ethernet frame only carries one high-level protocol "letter" at a time and not a whole bag full, but you get the idea. The Ethernet frame is then placed on the Ethernet media system for delivery to Station B.

Protocol encapsulation

The independent system of the high-level protocol packet has its own addresses and data. The Ethernet frame has its own data field that is used for carrying the high-level protocol packet. This kind of organization is called encapsulation, and it's a common theme in the networking world. Encapsulation is the mechanism that allows independent systems to work together, such as network protocols and Ethernet LANs. With encapsulation the Ethernet frame carries the network protocol packet by treating the entire packet as just so much unknown data stuffed into the data field of the Ethernet frame. Upon delivery of the Ethernet frame to the station, it's up to the network software running on the station to deal with the protocol packet extracted from the Ethernet frame's data field.

Just like a trucking system carrying packages, the Ethernet system is fundamentally unaware of what is packed inside the high-level protocol packets that it carries between computers. This allows the Ethernet system to carry all manner of network protocols without worrying about how each high-level protocol works. In order to get the network protocol packet to its destination, however, the protocol software and the Ethernet system must interact to provide the correct destination address for the Ethernet frame. When using TCP/IP, the destination address of the IP packet is used to discover the Ethernet destination address of the station for which the packet is intended. Let's look briefly at how this works.

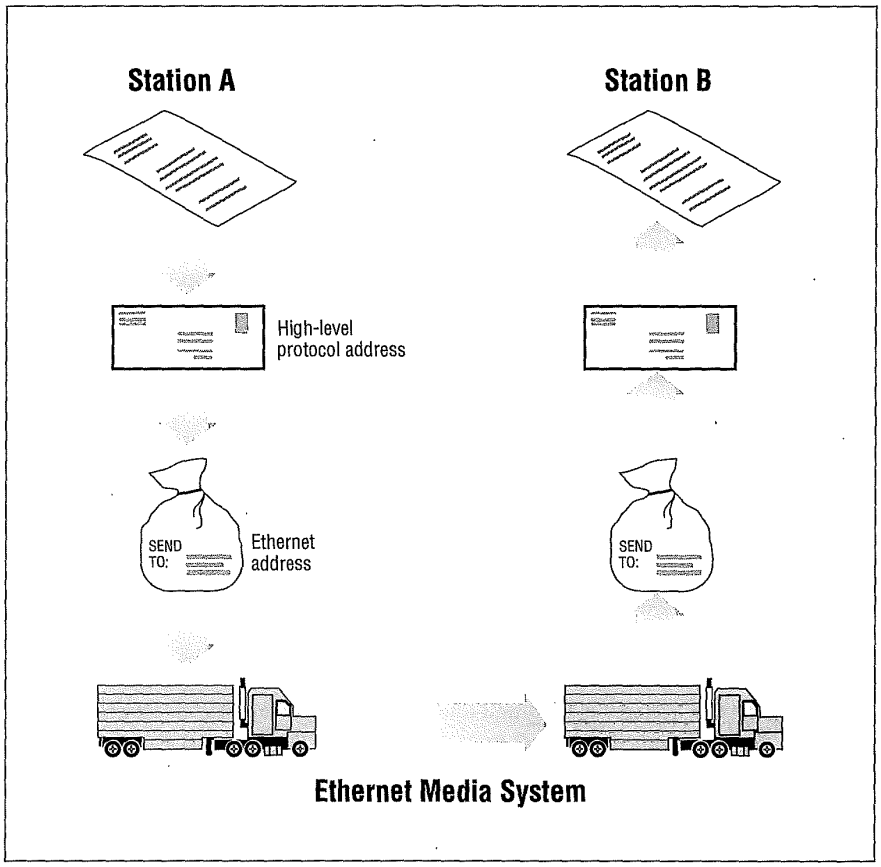


Figure 2-4. Ethernet and network protocols

Internet Protocol and Ethernet Addresses

High-level network protocols have their own system of addresses, such as the 32-bit address used by the current version of the Internet Protocol (IPv4) used on the Internet. The IP-based networking software in a given computer is aware of the 32-bit IP address assigned to that computer and can also read the 48-bit Ethernet address of its network interface. However, it doesn't know what the Ethernet addresses of the other stations on the network are when first trying to send a packet over the Ethernet.

To make things work, there needs to be a way to discover the Ethernet addresses of other IP-based computers on the network. The TCP/IP network protocol system accomplishes this task by using the Address Resolution Protocol (ARP).

Operation of the ARP protocol

The ARP protocol is fairly straightforward. Figure 2-5 shows two stations—Station A and Station B—sending and receiving an ARP packet over an Ethernet.

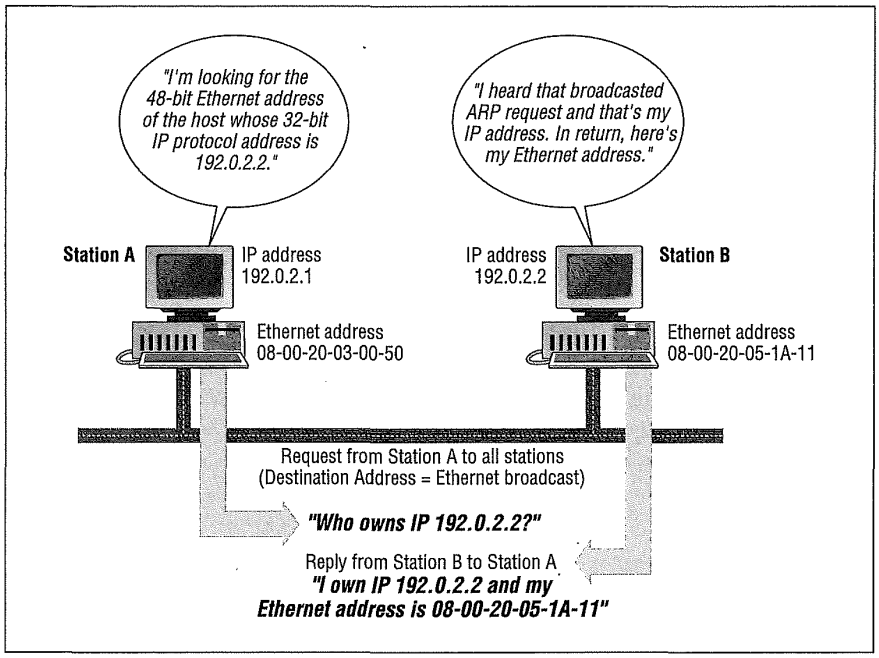


Figure 2-5. Using ARP over an Ethernet

Station A has been assigned the 32-bit IP address of 192.0.2.1, and wishes to send data over the Ethernet channel to Station B, which has been assigned IP address 192.0.2.2. Station A sends a packet to the broadcast address containing an ARP request. The ARP request basically says, "Will the station on this Ethernet that has the IP address of 192.0.2.2 please tell me what the 48-bit hardware address of their Ethernet interface is?"

Since the ARP request is sent in a broadcast frame, every Ethernet interface on the network reads it and hands the ARP request to the networking software running on the station.* Only Station B with IP address 192.0.2.2 will respond, sending a

* It would be a better idea to use a specified multicast address for this purpose; so that only IP-speaking computers would receive IP ARPs, and computers running other protocols would not be bothered. The address resolution system in the new version of IP (IPv6) uses multicast addressing for this reason. As of this writing IPv4 is the most widely used network protocol, although it is expected that a conversion to IPv6 will occur eventually.

packet containing the Ethernet address of Station B back to the requesting station. Now Station A has an Ethernet address to which it can send frames containing data destined for Station B, and the protocol communication can proceed.

That's all there is to it! The ARP protocol provides the "glue" between the 32-bit addresses used by the IP network protocols and the 48-bit addresses used by the Ethernet interfaces. The computers involved build a table in memory called the ARP cache to hold the IP addresses and their associated Ethernet addresses. Once this table is created by the ARP protocol, the network software can then look up the IP addresses in the table and find which Ethernet address to use when sending data to a given IP-based machine on the network.

In this chapter:

- *The Ethernet Frame*
- *Media Access Control Rules*
- *Essential Media System Timing*
- *Collision Detection and Backoff*
- *Gigabit Ethernet Half-Duplex Operation*
- *Collision Domain*
- *Ethernet Channel Capture*
- *High-level Protocols and the Ethernet Frame*

3

The Media Access Control Protocol

The tutorial in Chapter 2, *The Ethernet System*, introduced the Ethernet system and provided a brief look at how it works. In this chapter we take a much more detailed look at the original mode of operation used for Ethernet, which is based on the CSMA/CD media access control (MAC) protocol. This is also called half-duplex mode, to distinguish it from the full-duplex mode of operation. Full-duplex mode, which has no need for a MAC protocol, is described in Chapter 4, *Full-Duplex Ethernet.*

In the original half-duplex mode, the MAC protocol allows a set of stations to compete for access to a shared Ethernet channel in a fair and equitable manner. The protocol's rules determine the behavior of Ethernet stations, including when they are allowed to transmit a frame onto a shared Ethernet channel and what to do when a collision occurs.

Since there is no central controller in an Ethernet system, each Ethernet interface operates independently while using the same MAC protocol. By equipping all interfaces with the same set of rules, all stations connected to a shared Ethernet channel operate the same way. Therefore, the MAC protocol functions as a kind of "Robert's Rules for Robots."

You don't need to know all the details of the MAC protocol in order to build and use Ethernet LANs. However, an understanding of the MAC protocol can certainly help when designing networks or troubleshooting problems. The media guidelines

and MAC protocol are not some arbitrary laws dreamed up by a standards committee, but instead arise from the basic design and operation of the Ethernet system.

To simplify the description of the MAC protocol, this chapter has been broken down into three parts. The first part looks at the structure of the frame and the media access control rules used for transmitting frames, as well as how those rules affect the design of an Ethernet LAN. The next section takes a closer look at the operation of the important collision detect mechanism. Collision detection in Ethernet is frequently misunderstood, and we'll explain how it works, and how to design networks so that the collision detect system can do its job correctly. The last portion of the chapter describes how the high-level network software on your computer uses Ethernet frames to send data.

The Ethernet Frame

The Ethernet specifications determine both the structure of a frame and when a station is allowed to send a frame. Ethernet media access control is based on Carrier Sense with Multiple Access and Collision Detect, which gives rise to the CSMA/CD acronym described in Chapter 2. The Ethernet system exists to move frames carrying application data between computers; the organization of the frame is central to the operation of the system.

The frame was first defined in the original Ethernet DEC-Intel-Xerox (DIX) standard, and was later redefined in the IEEE 802.3 standard, which is now the official Ethernet standard. The changes between the two standards were mostly cosmetic, except for the type field.

The DIX standard defined a type field in the frame. The first 802.3 standard (published in 1985) specified this field as a length field, with a mechanism that allowed both versions of frames to coexist on the same Ethernet. As it happens, most networking software kept using the type field version of the frame, and the IEEE 802.3 standard was recently changed to define this field as being either length or type, depending on usage.

Figure 3-1 shows the DIX and IEEE versions of the Ethernet frame. Since DIX and IEEE frames are identical in terms of the number and length of fields, Ethernet interfaces can be used to send either kind of frame. The only difference in the frames is in the contents of the fields and the subsequent interpretation of those contents by the stations that send and receive the frames. Next, we'll take a detailed tour of the frame fields. This tour will describe what each frame field does and what the differences are between the two versions of the frame.

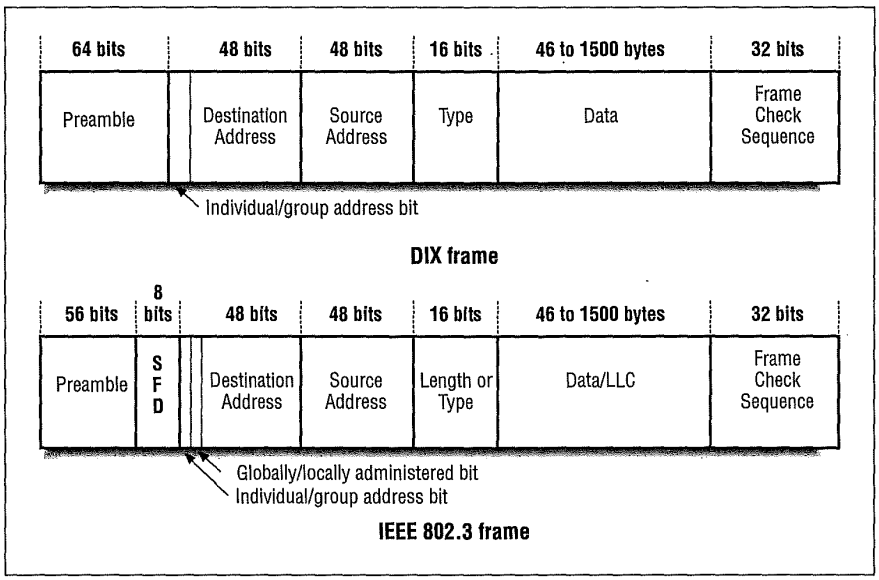


Figure 3-1. DIX Ethernet and IEEE 802.3 frames

Preamble

The frame begins with the 64-bit preamble field, which allows 10 Mbps Ethernet interfaces on the network to synchronize themselves with the incoming data stream before the important data fields arrive. The preamble exists in order to allow the beginning of the frame to lose a few bits due to signal start-up delays as it travels through a 10 Mbps system. This protects the rest of the frame from these effects. Like the heat shield of a spacecraft, which protects the spacecraft from burning up during re-entry, the preamble is the shield that protects the bits in the rest of the frame.

The preamble is maintained in the Fast Ethernet and Gigabit Ethernet systems to provide compatibility with the original Ethernet frame. However, both Fast Ethernet and Gigabit Ethernet systems use more complex mechanisms for encoding the signals that avoid any signal start-up losses. These two systems don't need the preamble to protect the frame signals, and it is preserved only for backward compatibility. The following discusses how the different standards implement the preamble:

DIX standard

In the DIX standard the preamble consists of eight "octets," or 8-bit bytes, of alternating ones and zeroes. The eighth byte of the preamble contains 6 bits of alternating ones and zeroes, but ends with the special pattern of "1, 1." These

two bits signal the receiving interface that the end of the preamble has been reached, and that the bits that follow are actual fields of the frame.

IEEE standard

In the 802.3 specification, the preamble field is formally divided into two parts consisting of seven bytes of preamble and one byte, called the *start frame delimiter* (SFD). The last two bits of the SFD are 1, 1—as with the DIX Standard. Although the IEEE standard chooses to define an SFD field that includes the last eight bits of the preamble, there is no practical difference between the DIX and IEEE preambles. The pattern of bits being sent is identical.

Destination Address

The destination address field follows the preamble. Each Ethernet interface is assigned a unique 48-bit address, called the interface's physical or hardware address. The destination address field contains the 48-bit Ethernet address that corresponds to the address of the interface in the station that is the destination of the frame. The destination address field may instead contain a 48-bit multicast address that one or more interfaces on the network have been enabled to receive, or the standard broadcast address.

Every Ethernet interface attached to the network reads in every transmitted frame up to at least the destination address field. If the destination address does not match the interface's own Ethernet address or one of the multicast or broadcast addresses that the interface is programmed to receive, then the interface is free to ignore the rest of the frame. The following discusses how the different standards implement destination addresses:

DIX standard

The first bit of the address, as sent onto the network medium, is used to distinguish physical addresses from multicast addresses. If the first bit is zero, then the address is the *physical address* of an interface, which is also known as a *unicast address* since a frame sent to this address only goes to one destination. If the first bit of the address is a one, then the frame is being sent to a *multicast address.*

IEEE standard

The IEEE 802.3 version of the frame adds significance to the second bit of the destination address, which is used to distinguish between *locally* and *globally* administered addresses. A globally administered address is a physical address assigned to the interface by the manufacturer, which is indicated by setting the second bit to zero. (DIX Ethernet addresses are always globally administered.) If the address of the Ethernet interface is administered locally for some reason,

then the second bit is supposed to be set to a value of one.* In the case of a broadcast address, the second bit is also a one in both the DIX and IEEE standard.

Understanding Physical Addresses

In Ethernet, the 48-bit physical address is written as 12 hexadecimal digits with the digits paired in groups of two, representing an octet (8 bits) of information. The octet order of transmission on the Ethernet is from the leftmost octet (as written or displayed) to the rightmost octet. The actual transmission order within the octet, however, is starting from the least significant bit of the octet through to the most significant bit. This means that an Ethernet address that is written as the hexadecimal string F0-2E-15-6C-77-9B is equivalent to the following sequence of bits, sent over the Ethernet channel from left to right:

```
0000 1111 0111 0100 1010 1000 0011 0110 1110 1110 1101 1001
```

Therefore, the 48-bit destination address that begins with the hexadecimal value 0xF0 is not a multicast address, since the first bit sent on the channel is a zero.

Source Address

The next field in the frame is the *source address*. This is the physical address of the interface that sent the frame. The source address is not interpreted in any way by the Ethernet MAC protocol. Instead, it is provided for the use of high-level protocols. An Ethernet station uses its physical address as the source address in any frame it transmits.

The DIX standard notes that a station can change the Ethernet source address, while the IEEE standard does not specifically state that an interface may have the ability to override the 48-bit physical address assigned by the manufacturer. However, all Ethernet interfaces in use these days appear to allow the physical address to be changed, which makes it possible for the network administrator or the high-level network software to modify the Ethernet interface address if necessary.

* Locally administered addresses are very rarely used on Ethernet systems, since the vast majority of Ethernet interfaces are assigned their own unique 48-bit address. Locally administered addresses, however, have been commonly used on some Token Ring systems.

To provide the physical address used in the source address field, a vendor of Ethernet equipment acquires an Organizationally Unique Identifier (OUI) from the IEEE. This unique 24-bit identifier is assigned by the IEEE. The OUI forms the first half of the physical address of any Ethernet interface that the vendor manufactures. As each interface is manufactured, the vendor also assigns a unique address to the interface using the lower 24 bits of the 48-bit address space. This results in a 48-bit physical address for each interface that contains an OUI in the first half of the address. The OUI can often be used to identify the vendor, which can be helpful when troubleshooting network problems.

VLAN Tag Header

Figure 3-1 shows what the typical Ethernet frame used by the vast majority of Ethernet devices looks like. However, a four-byte-long virtual LAN (VLAN) *tag header* may optionally be inserted in an Ethernet frame between the source address and the Length/Type field to identify the VLAN to which the frame belongs. VLANs are used in switching hubs as a way to direct Ethernet traffic to those ports of the hub which are defined to be members of a given VLAN. VLAN traffic may be sent between switching hubs and other devices by using the tag header to tag the Ethernet frames with a VLAN identifier. Until recently, VLAN tagging had been accomplished using a variety of proprietary approaches. Development of the IEEE 802.1Q standard for virtual bridged LANs provides the VLAN tag header as a vendor-neutral mechanism for identifying which VLAN a frame belongs to.

The addition of the four-byte VLAN tag header causes the maximum size of an Ethernet frame to be extended from the old maximum of 1518 bytes (not including the preamble) to a new maximum of 1522 bytes. Since VLAN tag headers are only added to Ethernet frames by switching hubs and other devices that have been programmed to send and receive VLAN-tagged frames, this does not affect traditional Ethernet operation.* VLANs and the contents and organization of VLAN tag headers are described in Chapter 18, *Ethernet Switching Hubs.*

* The first two bytes of the VLAN tag header contain a valid Ethernet type identifier. Therefore, if an Ethernet station that is not programmed to send or receive a VLAN tagged frame happens to receive a tagged frame, it will see what looks like a type identifier for an unknown protocol type and discard the frame. Even though a VLAN-tagged frame exceeds the original Ethernet maximum frame size of 1518 by four bytes, this is also not a problem. While developing the VLAN tag header specification, the IEEE engineers checked every Ethernet interface chip they could find and ascertained that they all could receive a 1522-byte frame without difficulty.

Type Field or Length Field

The next field in the Ethernet frame is either a type field or a length field. The following discusses how the different standards implement the type and/or length fields:

DIX standard

In the DIX Ethernet standard, this 16-bit field is called a *type field*, and always contains an identifier that refers to the type of high-level protocol data being carried in the data field of the Ethernet frame. For example, the hexadecimal value 0x0800 has been assigned as the identifier for the Internet Protocol (IP). A DIX frame being used to carry an IP packet is sent with the value of 0x0800 in the type field of the frame.

IEEE standard

When the IEEE 802.3 standard was first published in 1985, the type field was not included and instead the IEEE specifications called this field a *length field.* Type fields were added to the IEEE 802.3 standard in 1997, so the use of a type field in the frame is now officially recognized in 802.3. The identifiers used in the type field were originally assigned and maintained by Xerox, but now that the type field is part of the IEEE standard the responsibility for assigning type numbers has been transferred to the IEEE. In the most recent IEEE 802.3 standard this field is now called a Length/Type field, and the hexadecimal value in the field indicates the manner in which the field is being used.

If the value in this field is numerically equal to or less than the maximum untagged frame size in octets of 1518 (decimal), then the field is being used as a length field. In that case, the value in the field indicates the number of logical link control (LLC) data octets that follow in the data field of the frame. If the number of LLC octets is less than the minimum required for the data field of the frame, then octets of pad data will automatically be added to make the data field large enough. The content of pad data is unspecified by the standard. Upon reception of the frame, the length field is used to determine the length of valid data in the data field, and the pad data is discarded.

If the value in this field of the frame is numerically greater than or equal to 1536 decimal (0x600 hex), then the field is being used as a type field as specified in the original DIX standard. In that case, the hexadecimal identifier in the field is used to indicate the type of protocol data being carried in the data field of the frame. In the DIX standard the network software on the station is responsible for providing any pad data required to ensure that the data field is 46 bytes in length.

Data Field

Next comes the data field of the frame. The following discusses how the different standards implement the data field:

DIX standard

In a DIX frame this field must contain a minimum of 46 bytes of data, and may range up to a maximum of 1500 bytes of data. The network protocol software is expected to provide at least 46 bytes of data.

IEEE standard

The total data field of the IEEE 802.3 frame is the same length as the DIX frame: a minimum of 46 bytes and a maximum of 1500. However, a logical link control (LLC) protocol defined in the IEEE 802.2 LLC standard may ride in the data field of the 802.3 frame to provide control information. The LLC protocol is also used as a way to identify the type of protocol data being carried by the frame if the type/length field is used for length information. The LLC PDU is carried in the first set of bytes in the data field of the IEEE frame. The structure of the LLC PDU is defined in the IEEE 802.2 LLC standard.

The process of figuring out which protocol software stack gets the data in an incoming frame is known as *demultiplexing*. An Ethernet frame may use the type field to identify the high-level protocol data being carried by the frame. In the LLC specification, the receiving station demultiplexes the frame by deciphering the contents of the logical link control protocol data unit (PDU). These issues are described in more detail later in this chapter, in the section entitled "Multiplexing Data in Frames."

FCS Field

The last field in both the DIX and IEEE frame is the frame check sequence (FCS) field, which is also called the cyclic redundancy check, or CRC. This 32-bit field contains a value that is used to check the integrity of the various bits in the frame fields (not including preamble/SFD). This value is computed using the CRC, which is a polynomial that is calculated using the contents of the destination, source, type (or length), and data fields. As the frame is generated by the transmitting station, the CRC value is simultaneously being calculated. The 32 bits of the CRC value that are the result of this calculation are placed in the FCS field as the frame is sent. The x^{31} coefficient of the CRC polynomial is sent as the first bit of the field and the x^0 coefficient as the last bit.

The CRC is calculated again by the interface in the receiving station as the frame is read in. The result of this second calculation is compared with the value sent in the FCS field by the originating station. If the two values are identical, then the

receiving station is provided with a high level of assurance that no errors have occurred during transmission over the Ethernet channel.

End of Frame Detection

The presence of a signal on the Ethernet channel is known as *carrier*. The transmitting interface stops sending data after the last bit of a frame is transmitted, which causes the Ethernet channel to become idle. In the 10 Mbps system, the loss of carrier when the channel goes idle signals the receiving interface that the frame has ended. When the interface detects loss of carrier, it knows that the frame transmission has come to an end.

The carrier sense signal may continue to be asserted by the transceiver for a short period beyond the end of the frame due to some delay in the electronics of the transceiver. If that happens, it's possible for the interface to read in some random noise as being one or more phantom bits past the actual end of the frame. These extra bits are called *dribble bits*.

The standard notes that the interface should truncate the received frame to the nearest octet boundary after the end of carrier sense, which discards the dribble bits (as long as there are fewer than eight dribble bits). If the CRC value in the received frame is correct, then the frame is accepted as valid. This mechanism allows the interface to throw away up to seven dribble bits that may appear at the end of a frame.

The Fast Ethernet and Gigabit Ethernet systems use more complex signal encoding schemes, which have special symbols available for signaling the start and end of a frame. In these systems, the interface detects the beginning and end of a frame by way of the special symbols that precede and follow a frame transmission. These signal encoding schemes are described in greater detail in Chapter 6, *Ethernet Media Fundamentals*.

Media Access Control Rules

Now that we've seen what the structure of a frame is, let's look at the rules used for transmitting a frame on a half-duplex shared Ethernet channel. When transmitting a frame the station goes through the following steps:

- When a signal is being transmitted on the channel, that condition is called *carrier*.
- When a station attached to an Ethernet wants to transmit a frame, it waits until the channel goes idle, as indicated by an *absence of carrier*.

- When the channel becomes idle, the station waits for a brief period called the *interframe gap* (IFG), and then transmits its frame.
- If two stations happen to transmit simultaneously, they detect the collision of signals and reschedule their frame transmission. This occurrence is referred to as *collision-detect.*

There are two major things an interface connected to a half-duplex channel must do when it wants to send a frame. It must figure out when it can transmit, and it must be able to detect and respond to a collision. We'll first look at how the interface figures out when to transmit, and then we'll describe the collision detect mechanism.

The rules governing when an interface may transmit a frame are simple:

1. If there is no carrier (i.e., the medium is idle) and the period of no carrier has continued for an amount of time that equals or exceeds the IFG, then transmit the frame immediately. If a station wishes to transmit multiple frames, it must wait for a period equal to the IFG between each frame.

   The IFG is provided to allow a very brief recovery time between frame reception for the Ethernet interfaces. IFG timing is set to 96 bit times. That is 9.6 microseconds (millionths of a second) for the 10 Mbps varieties of Ethernet, 960 nanoseconds (billionths of a second) for the 100 Mbps varieties of Ethernet, and 96 nanoseconds for Gigabit Ethernet.

2. If there is carrier (i.e., the channel is busy), then the station continues to listen until the carrier ceases (i.e., the channel is idle). This is known as *deferring* to the passing traffic. As soon as the channel becomes idle, the station may begin the process of transmitting a frame, which includes waiting for the interframe gap interval.

3. If a collision is detected during the transmission, the station will continue to transmit 32 bits of data (called the *collision enforcement jam signal*). If the collision is detected very early in the frame transmission, then the station will continue sending until it has completed the preamble of the frame, after which it will send the 32 bits of jam.

   Completely sending the preamble and transmitting a jam sequence guarantees that the signal stays on the media system long enough for all transmitting stations involved in a collision to recognize the collision and react accordingly.

   a. After sending the jam signal, the station waits a period of time chosen with the help of a random number generator and then proceeds to transmit again, starting over at step 1. This process is called *backoff.* The randomly chosen time makes it possible for colliding stations to choose

different delay times, so they will not be likely to collide with one another again.

b. If the next attempt to transmit the frame results in another collision, then the station goes through the backoff procedure again, but this time the range of backoff times that are used in the random choice process will increase. This reduces the likelihood of another collision and provides an automatic adjustment mechanism for heavy traffic loads.

4. Once a 10 Mbps or 100 Mbps station has transmitted 512 bits of a frame (not including the preamble) without a collision, then the station is said to have acquired the channel. On a properly functioning Ethernet, there should not be a collision after channel acquisition. The 512-bit time value is known as the *slot time* of the Ethernet channel. Gigabit Ethernet extended the slot time for half-duplex Gigabit Ethernet channels. The Gigabit Ethernet extensions and half-duplex mode timing issues are described later in this chapter.

   Once a station acquires the channel and transmits its frame, it also clears its collision counter, which was used to generate the backoff time. If it encounters a collision on the next frame transmission, it will start the backoff calculations anew.

Note that stations transmit their data one frame at a time, and that every station must use this same set of rules to access the shared Ethernet channel for each frame transmitted. This process ensures fair access to the channel for all stations, since all stations must contend equally for the next frame transmission opportunity after every frame transmission. The MAC protocol ensures that every station on the network gets a fair chance to use the network.

A half-duplex Ethernet operates as a logical signal bus in which all stations share a common signal channel. Any station can use the MAC rules to attempt to transmit whenever it wants to, since there is no central controller. However, for this process to work correctly, each station must be able to accurately monitor the condition of the shared channel. Most importantly, all stations must be able to hear the carrier caused by each frame transmission. In addition, the media system of an Ethernet must be configured to allow a station to receive news of a collision within a carefully specified amount of time, known as the slot time.

The slot time is based on the *maximum* round-trip signal propagation time of an Ethernet system. The actual round-trip propagation time for a given network system will vary, depending on the length and type of cabling in use, the number of devices in the signal path, and so on. The standard provides specifications for the maximum cable lengths and the maximum number of repeaters that can be used for each media variety. This ensures that the total round-trip time for any given

Ethernet built according to the standard will not exceed the maximum round-trip time incorporated in the slot time.

Essential Media System Timing

While signals travel very fast on an Ethernet, they still take a finite amount of time to propagate over the entire media system. The longer the cables used in the media system, the more time it takes for signals to travel from one end of the system to the other. The total round-trip time used in the slot time includes the time it takes for frame signals to go through all of the cable segments. It also includes the time it takes to go through all other devices, such as transceiver cables, transceivers and repeaters.

The maximum length for cable segments are carefully designed so that the essential signal timing of the system is preserved, even if you use maximum-length segments everywhere and the system is the largest allowed. The guidelines for each media variety incorporate the essential timing and round-trip signal delay requirements needed to make any half-duplex Ethernet up to the maximum-size system work properly. The cable segment guidelines are described in the media system chapters located in Part II, *Ethernet Media Systems*. The correct signal timing is essential to the operation of the MAC protocol, so let's look at the slot time in more detail.

Ethernet Slot Time

The total set of round-trip signal delays are summed up in the Ethernet slot time, which is defined as a combination of two elements:

- The time it takes for a signal to travel from one end of a maximum-sized system to the other end and return. This is called the *physical-layer round-trip propagation time.*
- The maximum time required by collision enforcement, which is the time required to detect a collision and to send the collision enforcement jam sequence.

Both elements are calculated in terms of the number of bit times required. Adding the two elements together plus a few extra bits for a fudge factor gives us a slot time that is 512 bit times for 10 and 100 Mbps systems. The Gigabit Ethernet slot time is described later in this chapter.

The time it takes to transmit a frame that is 512 bits long is slightly longer than the actual amount of time it takes for the signals to get to one end of a maximum size Ethernet and back. This includes the time required to transmit the jam sequence. Therefore, when transmitting the smallest legal frame, a transmitting station will

always have enough time to get the news if a collision occurs, even if the colliding station is at the other end of a maximum-sized Ethernet.

The slot time includes the signal propagation time through the maximum set of components that can be used to build a maximum length network. If you use media segments longer than those specified in the standard, the result will be an increase in the round-trip time, and this could end up adversely affecting the operation of the entire system.

Any components or devices that add too much signal delay to the system can have the same negative effect. Smaller networks, on the other hand, will have smaller round-trip times, which means collision detection will occur faster and collision fragments will be smaller.

Slot Time and Network Diameter

The maximum network cable length and the slot time are tightly coupled. The 512 bit times were chosen as a trade-off between maximum cable distance and a minimum frame size. The total cable length allowed in a network system determines the maximum diameter of that system. The following discusses slot time and network diameter for the three types of Ethernet systems:

Original 10 Mbps slot time

In the original 10 Mbps system, signals could travel roughly 2800 meters (9,186 feet) of coaxial cable and back in 512 bit times, which provided a nice long network diameter.*

Fast Ethernet slot time

When the Fast Ethernet standard was developed in 1995, the slot time was kept at 512 bits, since changing the minimum frame length would have required changes in network protocol software, network interface drivers, and Ethernet switches.

However, signals operate ten times faster in Fast Ethernet, which means that a Fast Ethernet bit time is ten times smaller than a bit time in the 10 Mbps original Ethernet system. Since each bit time is ten times smaller, that means it is "on the wire" for only one-tenth of the time. As a result, 512 bits will travel over approximately one-tenth of the cable distance with a Fast Ethernet system, compared to original Ethernet. This results in a maximum network diameter of roughly 205 meters (672.5 feet) in Fast Ethernet.

This was considered acceptable, since by 1995 most sites were using twisted-pair cabling. The twisted-pair structured cabling standards limit segments to a

* The much shorter 100 m (328 feet) target length for 10BASE-T twisted-pair segments is based on signal quality limitations, and not round-trip timing.

maximum of 100 meters (328 feet), which meant that the smaller maximum diameter of a half-duplex Fast Ethernet system was not a major hardship. In Chapter 18 we'll show how twisted-pair segments can be connected to switching hubs, and you'll see how multiple LANs can be combined without exceeding a 205 meter network diameter per LAN.

Gigabit Ethernet slot time

When Gigabit Ethernet was developed, keeping the slot time at 512 bits would result in a maximum network diameter of roughly 20 meters (65.6 feet) for half-duplex operation. That's too short, since 20 meters won't reach very far in a typical building. However, all the good reasons for keeping the minimum frame size at 512 bits still existed.

In response to this dilemma, the Gigabit Ethernet standard came up with a mechanism that maintains the minimum frame size at 512 bits, while extending the slot time to 4096 bits (512 bytes). This is done with a system called *carrier extension*, which is described later in this chapter.

Use of the Slot Time

By using the slot time as a basic parameter in the media access calculations, the designers can guarantee that an Ethernet system will work properly under all possible legal combinations of standard network components and cable segments.

The slot time is used in several ways:

- The slot time establishes the maximum upper bound for a station to acquire the shared network channel. Once a station has transmitted a frame for 512 bit times (that is long enough for every station on a maximum-sized Ethernet to have heard it, and long enough for any news of a collision to have returned from the farthest end of the network back to the transmitting station), it is assured it has acquired the channel. Assuming that there has not been a collision, after 512 bit times all other stations will have sensed carrier and will defer to the carrier signal. The transmitting station can now expect to transmit the rest of the frame without a collision. The slot time sets the upper bound on 10 Mbps and Fast Ethernet channel acquisition to 512 bit times.
- The 512 bits of the slot time also serve as the basic unit of time used by the backoff algorithm to generate a waiting period after a collision has occurred. This algorithm is described later in this chapter.
- A valid collision can only occur within the 512-bit slot time, because once all stations on a network have seen the signal on the medium (*carrier*), they will defer to carrier and won't transmit. Since a valid collision can only happen within the first 512 bits of frame transmission, this also establishes an upper bound on the length of the frame fragment that may result from a collision.

Based on this, the Ethernet interface can detect and discard frame fragments generated by collisions as the fragments are smaller than 512 bits and are too short to be valid frames.

Slot Time and Minimum Frame Length

Setting the minimum frame length at 512 bits (not including preamble) meant requiring that the data field must always carry at least 46 bytes. A frame that is carrying 46 bytes of data will be 512 bits long, and therefore will not be regarded as a collision fragment. The 512 bits include 12 bytes of addresses, plus 2 bytes used in the type/length field, plus 46 bytes of data, plus 4 bytes of FCS. The preamble is not considered part of the actual frame in these calculations.

The requirement that each frame carry 46 bytes of data does not impose much overhead when you consider how the data field of a typical frame is used. For example, the minimum length for a typical set of IPv4 headers and TCP headers in a TCP/IP packet is 40 bytes, leaving 6 bytes to be provided by the application sending the TCP/IP packet. If the application only sends a single byte of data in the TCP/IP packet, then the data field needs to be "padded out" with 5 bytes of pad data to provide the minimum of 46 bytes. That's not a severe amount of overhead, and most applications will send enough data so that there is no pad data needed at all.

Collision Detection and Backoff

Collision detection and backoff is an important feature of the Ethernet MAC protocol. It's also a widely misunderstood and misrepresented feature. Let's clear up a couple of points right away:

- *Collisions are not errors.* Instead, collisions are a normal part of the operation of an Ethernet LAN. They are expected to happen, and collisions are handled quickly and automatically.
- *Collisions do not cause data corruption.* As we've just seen, when a collision occurs on a properly designed and implemented Ethernet, it will happen sometime in the first 512 bit times of transmission. Any frame transmission that encounters a collision is automatically resent by the transmitting station. Any frame less than 512 bits long is considered a collision fragment, and is automatically and silently discarded by all interfaces.

As noted in the last chapter, it's unfortunate that the original Ethernet design used the word "collision" for this aspect of the Ethernet MAC protocol. Despite the name, collisions are not a problem on an Ethernet. Instead, the collision detect

and backoff feature is a normal part of the operation of Ethernet, and results in fast and automatic rescheduling of transmissions.

The collisions counted by a typical Ethernet interface are the ones that occur while that interface is trying to transmit a frame. Collision rates can vary widely, depending on the traffic rate on the Ethernet channel and on the number of transmitting stations. Even on very lightly loaded networks, collisions will occur occasionally. Collision rates may range anywhere from fractions of a percent of the number of frame transmissions on up to a larger percentage on heavily loaded networks. Collision rates can be significantly higher for very heavily loaded networks, such as a network supporting high-speed computers.

In any case, the thing to worry about is the total traffic load on the network. Since the collision rate is simply a reflection of the normal functioning of an Ethernet, the rate of collisions seen by a given interface is not significant. Chapter 19, *Ethernet Performance*, provides more information about measuring the performance of an Ethernet channel.

The Ethernet MAC mechanism, with its collision detect and backoff system, was designed to make it possible for independent stations to compete for access to the LAN in a fair manner. It also provides a way for stations to automatically adjust their behavior in response to the load on the network.

Operation of Collision Detect

Although the stations must listen to the network and defer to traffic (*carrier sense*), it is possible for two or more stations to detect an idle channel at the same instant and transmit simultaneously. A collision may occur during the initial part of a station's transmission, which is the 512-bit slot time, also called the *collision window*. The collision window lasts for the amount of time it takes for the signal from a station to propagate to all parts of the shared channel and back. Once the collision window has passed, the station is said to have acquired the channel. No further opportunity for collision should exist, since all other stations can be assumed to have noticed the signal (via carrier sense) and to be deferring to its presence.

Collision detection on media systems

The actual method used to detect a collision is medium-dependent. A link segment medium, such as twisted-pair or fiber optic Ethernet, has independent transmit and receive data paths. A collision is detected in a link segment transceiver by the simultaneous occurrence of activity on both transmit and receive data paths.

On a coaxial cable medium, the transceivers detect a collision by monitoring the average DC signal level on the coax. When two or more stations are transmitting simultaneously, the average DC voltage on the coax reaches a level that triggers

Collision Propaganda

The Ethernet collision algorithm is one of the least understood and most widely misrepresented parts of the entire system. You will sometimes see articles in networking magazines or trade journals that imply that Ethernets are littered with collided frames, or that the collision detection and resolution mechanism sets severe limits on the throughput of the system. Instead, the Ethernet collision detect mechanism is a normal part of the operation of an Ethernet.

If you find yourself reading such an article in which the word *collision* is presented as some sort of problem, try substituting the phrase *channel arbitration sensing* for each occurrence of the word "collision" in what you're reading. This might help clear up any misunderstandings, and may also reveal the cases in which a writer does not understand what collision detect really does in Ethernet. The collision detect and backoff mechanism is simply a fast and low-overhead way to resolve any simultaneous transmissions that occur on a network system that allows multiple access to the channel.

Those who claim collisions are a problem on Ethernet are usually unaware of how the system really works, or may have read reports that misrepresent the workings of the Ethernet system. Some of these reports have led to persistent myths about Ethernet performance that have no basis in fact. These myths are dealt with in more detail in Chapter 19. In reality, the collision backoff algorithm was designed to allow stations to respond automatically to varying traffic levels, allowing the stations to avoid one another's transmissions. The collision rate on a properly functioning Ethernet is simply a reflection of how busy the network is, and of how many stations are trying to access the channel.

the collision detect circuit in the coax transceiver. A coaxial transceiver continually monitors the average voltage level on the coaxial cable, and sends a collision detect signal to the Ethernet interface when the average voltage level indicates that multiple stations are simultaneously transmitting. This process takes slightly longer than collision detection on link segments; the increased time is included in the calculations used for total signal delay on a 10 Mbps Ethernet.

Late Collisions

Normal collisions occur during the first 512 bits of frame transmission. If a collision occurs after 512 bit times, then it is considered an error and called a *late collision.* A late collision is a serious error, since it indicates a problem with the network system, and since it causes the frame being transmitted to be discarded.

The Ethernet interface will not automatically retransmit a frame lost due to a late collision. This means that the application software must detect the lack of response

due to the lost frame, and retransmit the information. This can take a significant amount of time, caused by waiting for the acknowledgment timers in the application software to time out and resend the information. Therefore, even a small number of late collisions can result in slow network performance. Any report of late collisions by devices on your network should be taken seriously, and the problem should be resolved as soon as possible.

Common causes of late collision

The most common cause of late collisions is a mismatch between the duplex configuration at each end of a link segment. For example, late collisions will occur if a station at one end of a link is configured for half-duplex operation, and a switch port on the other end of the link is configured for full-duplex. Full-duplex shuts off the CSMA/CD protocol and a full-duplex port sends data whenever it wants to. If the full-duplex end of the link happens to transmit while the half-duplex end is sending a frame and has acquired the channel, then the half-duplex end sees a late collision.

Late collisions can also be caused by problems with the media, such as a twisted-pair segment with excessive signal crosstalk. A twisted-pair transceiver detects collision by the simultaneous occurrence of traffic on the transmit and receive signal paths. Therefore, the occurrence of excessive signal crosstalk between the transmit and receive wire pairs can cause the transceiver to detect "phantom collisions." If the crosstalk takes long enough to build up to levels that will trigger the collision detection circuit, a late collision can be generated.

The Collision Backoff Algorithm

After a collision is detected, the transmitting stations reschedule their transmissions using the backoff algorithm to minimize the chances of another collision. This algorithm also allows a set of stations on a shared Ethernet channel to automatically modify their behavior in response to the activity level on the network. The more stations you have on a network, and the busier they are, the more collisions there will be. In the event of multiple collisions for a given frame transmission attempt, the backoff algorithm provides a way for a given station to estimate how many other stations are trying to access the network at the same time. This allows the station to adjust its rate of retransmission accordingly.

When the transceiver attached to the Ethernet senses a collision on the medium, it sends a collision presence signal back to the station interface. If the collision is sensed very early in the frame transmission, the transmitting station interface does not respond to a collision presence signal until the preamble has been sent completely. At this point, it sends out 32 bits of jam signal and stops transmitting. As a result, the collision signal will stay long enough for all other transmitting stations to see it. The stations involved in transmitting frames at the time of the collision

must then reschedule their frames for retransmission. The transmitting stations do this by generating a period of time to wait before retransmission, which is based on a random number chosen by each station and used in that station's backoff calculations.

On busy Ethernet channels, another collision may occur when retransmission is attempted. If another collision is encountered, the backoff algorithm provides a mechanism for adjusting the timing of retransmissions to help avoid congestion. The scheduling process is designed to exponentially increase the range of backoff times in response to the number of collisions encountered during a given frame transmission.

The more collisions per frame, the larger the range of backoff times that will be generated for use by the station interface in rescheduling its next transmission attempt. This means that the more collisions that occur for a given frame transmission, the likelier it is that a station will wait longer before retrying. The name for this scheduling process is *truncated binary exponential backoff. Binary exponential* refers to the power of two used in the delay time selection process, while *truncated* refers to the limit set on the maximum size of the exponential.

Operation of the Backoff Algorithm

After a collision occurs, the basic delay time for a station to wait before retransmitting is set at a multiple of the 512 bit Ethernet slot time used in 10 and 100 Mbps Ethernet. Note that a bit time is different for each speed of Ethernet, occupying 100 nanoseconds on 10 Mbps Ethernet and 10 nanoseconds on Fast Ethernet.

The amount of total backoff delay is calculated by multiplying the slot time by a randomly chosen integer. The range of integers used in the choice is generated according to the rules of the backoff algorithm, which create a range of integers that increases in size after each collision that occurs for a given frame transmission attempt. The interface then randomly chooses an integer from this range, and the product of this integer and the slot time creates a new backoff time.

The backoff algorithm uses the following formula for determining the integer r, which is used to multiply the slot time and generate a backoff delay:

$$0 \leq r < 2^k$$

where $k = \min(n,10)$.

To translate into something closer to English:

- r is an integer randomly selected from a range of integers. The value of r may range from zero to one less than the value of two to the exponent, k.
- k is assigned a value that is equal to either the number of transmission attempts or the number 10, whichever is less.

Let's walk through the algorithm and see what happens. Imagine that an interface tries to send a frame and a collision occurs. On the first retry of this frame transmission, the value for 2^k is two since the number of tries is one. (Two to the exponent one—2^1—gives us a result of two.) The range of possible integers to choose from for the first retry is set by the equation to anything from zero or greater, but less than two.

Therefore, on the first retry after a collision, the interface is allowed to randomly choose a value of r from the range of integers of zero to one. The practical effect is that after the first collision on a given frame transmission, the interface may wait zero slot times (i.e., zero microseconds [μs]), and reschedule the next transmission immediately. Alternatively, the station may wait one slot time, for a backoff time of 51.2 μs on a 10 Mbps channel.

If the network is busy due to a number of other stations trying to transmit, and the frame retransmission attempt happens to result in another collision, we're then on try number two. Now the random number r is selected from a set of values that range from zero to one less than the value of two to the exponent of two. As a result, this time around the interface may randomly choose the slot time multiplier from the set of numbers 0, 1, 2, 3. The integer that is chosen is used to multiply the slot time and develop a backoff time before retransmitting.

Therefore, if the interface sees a collision on the second attempt to transmit the same frame, it will randomly choose to wait zero, one, two, or three times the slot time before retransmitting again. This expansion in the range of integers is the part of the algorithm that provides an automatic adjustment for heavy network traffic loads and the repeated collisions that they cause.

Choosing a Backoff Time

Notice that the interface is *not* required always to pick a larger integer and develop a larger backoff time after each collision on a given frame transmission. Instead, what happens is that the *range* of integers to choose from gets larger and the *probability* that the interface will generate a longer backoff time increases. If a collision happens again for that frame attempt, the process continues with the value of k (and hence, the range of the number r) exponentially increasing for the first 10 retries of a given frame transmission attempt.

After ten retries, the value of k stops increasing, which represents the truncation of the algorithm. At this point 2^k is equal to 1024, so that r is being chosen from the range of numbers from 0 through 1023. If the frame continues to encounter collisions after 16 retries, the interface will give up. The interface discards the frame, reports a transmission failure for that frame to the high-level software, and proceeds with the next frame, if any.

On a network with a reasonable load, a station will typically acquire the channel and transmit its frame within the first few tries. Once the station has done so, it clears its backoff counter. If a station encounters a collision on its next frame transmission attempt, it will calculate a new backoff time.

Since an idle channel is instantly available to the first station that wants to send a frame, the MAC protocol provides very low access time when the offered load on the Ethernet channel is light. Stations wishing to transmit frames will experience longer wait times as the load on the channel increases based on the retransmission times generated by the backoff algorithm.

In effect, each station makes an estimate of the number of other stations that are providing a load on the channel. The backoff algorithm provides a way for stations to do this by allowing the stations to base the estimate on the one property of the network traffic that each station can easily monitor. That property is the number of times that a station has tried to send a given frame and has detected a collision. Table 3-1 shows the estimates that a station makes and the range of backoff times that may occur on a 10 Mbps channel.

Table 3-1. Maximum Backoff Times on a 10 Mbps System

Collision on Attempt Number	Estimated Number of Other Stations	Range of Random Numbers	Range of Backoff Times[a]
1	1	0...1	0...51.2 μs
2	3	0...3	0...153.6 μs
3	7	0...7	0...358.4 μs
4	15	0...15	0...768 μs
5	31	0...31	0...1.59 ms
6	63	0...63	0...3.23 ms
7	127	0...127	0...6.50 ms
8	255	0...255	0...13.1 ms
9	511	0...511	0...26.2 ms
10–15	1023	0...1023	0...52.4 ms
16	too high	N/A	discard frame

[a] Backoff times are shown in microseconds (μs) and milliseconds (ms).

As Table 3-1 shows, a station exponentially increases its estimate of the number of other stations that are transmitting on the network. After the range reaches 1023, the exponential increase is stopped, or truncated. The truncation provides an upper limit on the backoff time that any station will need to deal with. The truncation has another effect, since it means that there are 1024 potential "slots" for a station to transmit. This number leads to the inherent limit of 1024 stations that can be supported by a half-duplex Ethernet system.

After 16 tries, the station gives up the frame transmission attempt and discards the frame. By the sixteenth attempt, the network is considered to be overloaded or broken—there is no point in retrying endlessly.

All of the numbers used in the collision and backoff mechanism were chosen as part of worst-case traffic and population calculations for an Ethernet system. The goal was to determine reasonable time expectations for a single station to wait for network access. On smaller Ethernets with smaller host populations, the stations detect collisions faster. This leads to smaller collision fragments and more rapid resolution of collisions among multiple stations.

Now that we've seen how the Ethernet half-duplex timing works for 10 and 100 Mbps Ethernet, we'll look next at half-duplex operation for Gigabit Ethernet.

Gigabit Ethernet Half-Duplex Operation

The Gigabit Ethernet half-duplex mode uses the same basic CSMA/CD access mechanism as the 10 and 100 Mbps varieties of Ethernet, with the major exception of the slot time. The slot time in Gigabit Ethernet was modified to accommodate the special timing constraints which arise from the speed of the system.

Currently, all Gigabit Ethernet equipment is based on the full-duplex mode of operation described in Chapter 4. To date, none of the vendors have plans to develop equipment based on half-duplex Gigabit Ethernet operation. Nonetheless, a half-duplex CSMA/CD mode for Gigabit Ethernet has been specified, if only to insure that Gigabit Ethernet met the requirements for inclusion in the IEEE 802.3 CSMA/CD standard. For the sake of completeness, a description of Gigabit Ethernet half-duplex mode is provided here.

Gigabit Ethernet Half-Duplex Network Diameter

A major challenge for the engineers writing the Gigabit Ethernet standard was to provide a sufficiently large network diameter in half-duplex mode. As we've seen, the maximum network diameter (i.e., cable distance) between any two stations largely determines the slot time, which is an essential part of the CSMA/CD MAC mechanism.

Repeaters, transceivers and the interfaces have circuits that require some number of bit times to operate. The combined set of these devices used on a network takes a significant number of bit times to handle frames, respond to collisions, and so on. It also takes a small amount of time for a signal to travel over a length of fiber optic or metallic cable. All of this results in the total timing budget for signal propagation through a system, which determines the maximum cabling diameter allowed when building a half-duplex Ethernet system.

In Gigabit Ethernet, the signaling happens ten times faster than it does in Fast Ethernet, resulting in a bit time that is one-tenth the size of the bit time in Fast Ethernet. Without any changes in the timing budget, the maximum network diameter of a Gigabit Ethernet system would be about one-tenth of that for Fast Ethernet, or in the neighborhood of 20 meters (65.6 feet).

Twenty meters is usable within a single room, such as a machine room equipped with a set of servers. However, one of the goals for the Gigabit Ethernet system is to support a large enough half-duplex network diameter to reach from a Gigabit Ethernet hub to the desktop in a standard office building. Desktop cabling for office buildings is typically based on structured cabling standards, which require the ability to reach 100 meters from a hub port. This means that the total network diameter may reach a maximum of 200 meters when connecting two stations to a Gigabit Ethernet repeater hub.*

Looking for Bit Times

To meet the 200 meter half-duplex network diameter goal, the designers of the Gigabit Ethernet system needed to increase the round-trip timing budget to accommodate longer cables. If it was somehow possible to speed up the internal operations of devices, such as repeaters, then it might be possible to increase the bit timing budget. The idea is that you could save some bit times that are consumed when signals are sent through repeater hubs. Unfortunately, with today's technology, it is not possible to produce repeaters and other components with ten times less delay than equivalent Fast Ethernet devices.

The next place you might think of looking to save bit times is in the cable propagation delays. However, it turns out that the signal propagation delay through the cables cannot be reduced, as the delay is fundamentally based on the speed of light (which is notoriously difficult to improve).

Another way to gain time and achieve longer cable distance was in the minimum frame transmission specification. If the minimum frame time was extended then the Ethernet signal would stay on the channel longer. This would extend the round-trip time of the system and make it possible to achieve the 200 meter goal for the cabling diameter over twisted-pair cable. The problem with this scheme, however, is that changing the minimum frame length would make the frame incompatible with the other varieties of Ethernet—all of which use the same standard minimum frame length.

* Repeaters can only be used in a half-duplex, shared Ethernet channel. By definition, anything connected to a repeater hub must be operating in half-duplex mode.

Carrier Extension

The solution to this conundrum is to extend the amount of time occupied by the signal carrier associated with a minimum frame transmission, without actually modifying the minimum frame length or other frame fields. Gigabit Ethernet does this by extending the amount of time a frame signal is active on a half-duplex system with a mechanism called carrier extension, as shown in Figure 3-2. The frame signal, or *carrier*, is extended by appending non-data signals, called extension bits. Extension bits are used when sending short frames so that the frame signal stays on the system for a minimum of 512 bytes (4,096 bit times), which is the new slot time for Gigabit Ethernet. This slot time makes it possible to use longer cables, and is also used in the collision backoff calculations on Gigabit Ethernet systems.

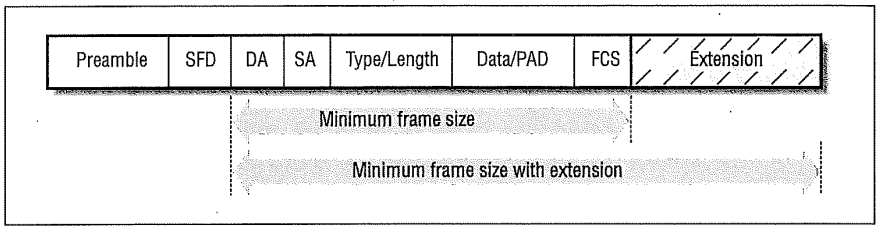


Figure 3-2. Carrier extension

The use of carrier extension bits assumes that the underlying physical signaling system is capable of sending and receiving non-data symbols. Signaling for all fiber optic and metallic cable Gigabit Ethernet systems is based on signal encoding schemes that provide non-data symbols that will trigger carrier detection in all station transceivers. This makes it possible to use these non-data symbols as carrier extension bits without having the extension bits confused with real frame data. These encoding schemes are described in more detail in Chapter 6.

With carrier extension, a minimum size frame of 64 bytes (512 bits) is sent on a Gigabit Ethernet channel along with 448 extension bytes (3,584 bits), resulting in a carrier signal that is 512 bytes in length. Any frame less than 4,096 bits long will be extended as much as necessary to provide carrier for 4,096 bit times (but no more).

Carrier extension is a simple scheme for extending the collision domain diameter; however, it adds considerable overhead when transmitting short frames. A minimum-size frame carrying 46 bytes of data is 64 bytes in length. Carrier extension adds another 448 bytes of non-data carrier extension bits when the frame is transmitted, which significantly reduces the channel efficiency when transmitting short frames.

The total impact on the channel efficiency of a network will depend on the mix of frame sizes seen on the network. As the size of the frame being sent grows, the number of extension bits needed during transmission of the frame is reduced. When the frame being sent is 512 bytes or more in length, no extension bits are used. Therefore, the amount of carrier extension overhead encountered when sending frames will vary depending on the frame size. Carrier extension is only used in half-duplex Gigabit Ethernet. In full-duplex mode the CSMA/CD MAC protocol is not used, which removes any concern about the slot time. Therefore, full-duplex Gigabit Ethernet links do not need carrier extension, and are able to operate at full efficiency.

Frame Bursting

The Gigabit Ethernet standard defines an optional capability called frame bursting to improve performance for short frames sent on half-duplex channels. This makes it possible for a station to send more than one frame during a given transmission event, improving the efficiency of the system for short frames. The total length of a frame burst is limited to 65,536 bit times plus the final frame transmission, which sets a limit on the maximum burst transmission time. Figure 3-3 shows how frame bursting is organized.

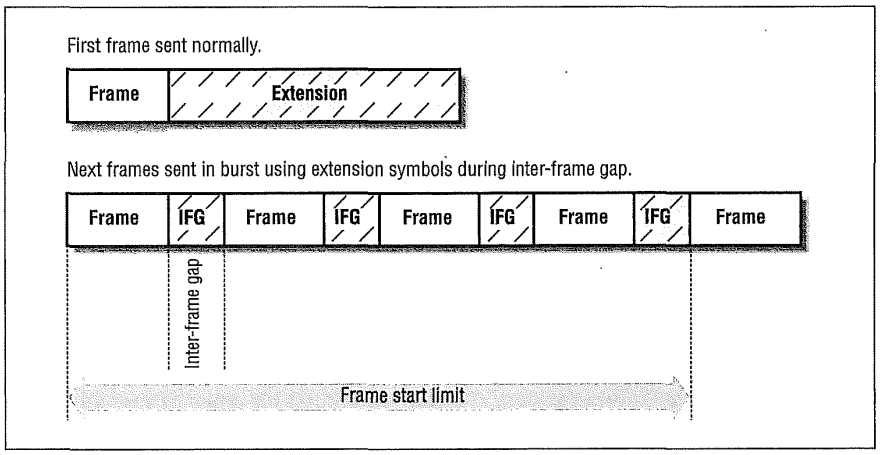


Figure 3-3. Frame bursting

Here's how the frame bursting system works. The first frame of the transmission is always sent normally, so that the first frame is sent by itself, and will be extended if necessary. Because collision always occurs within the first slot time, only this frame can be affected by a collision, requiring it to be retransmitted. This frame may encounter one or more collisions during the transmission attempt.

However, once the first frame (including any extension bits) is transmitted without a collision, then a station equipped with frame bursting can keep sending additional frames until the 65,536 bit time burst limit is reached. To accomplish this, the transmitting station must keep the channel from becoming idle between frame transmissions. If the station became idle during frame transmission, other stations would try to acquire the channel, leading to collisions.

The frame bursting station keeps the channel active by transmitting special symbols that are understood by all stations to be non-data symbols during the interframe gap times of the frames. This causes all other stations to continue to sense carrier (*activity*) on the channel. This, in turn, causes the other stations to continue deferring to passing traffic, allowing the frame bursting station to continue sending frames without concern that a collision will occur.

In essence, the first frame transmission clears the channel for the subsequent burst frames. Once the first frame has been successfully transmitted on a properly designed network, the rest of the frames in a burst are guaranteed not to encounter a collision. Frames sent within a burst do not require extension bits. The transmitting station is allowed to continue sending frames in a burst until the *Frame Burst Limit* (FBL) is reached, which is the time to the start of the last frame in the burst.

For short frames, the optional frame bursting mechanism can improve the utilization rate of the channel. However, this can only occur if the station software is designed to take advantage of the ability to send bursts of frames. Without frame bursting, a half-duplex Gigabit Ethernet channel is less than twice as fast as a Fast Ethernet channel for the shortest frame size. With frame bursting, the Gigabit Ethernet channel is slightly over nine times faster than the Fast Ethernet system for a constant stream of short frames.

Frame bursting and channel efficiency

Without frame bursting the channel efficiency is low for a Gigabit Ethernet channel carrying a constant stream of 64-byte frames (512 bits). It requires an overhead of one slot time to carry the 512 bit minimum size frame, which is 4096 bit times in the Gigabit Ethernet system. To this we add 64 bit times of preamble and 96 bit times of interframe gap, for a total of 4,256 bits of overhead. Dividing the 512 bit payload by the overhead of 4,256 bits reveals a 12 percent channel efficiency.

With frame bursting the Gigabit Ethernet channel is considerably more efficient for small frames, since a whole series of frames can be sent in a burst without overhead for the slot time once the channel is acquired. Theoretically, you could send 93 small frames in a single burst, with a channel efficiency of over 90 percent. However, it is unlikely that the smallest possible frames will dominate the traffic

flow in the real world. Nor is it likely that a given station will have so many small frames to send that it can pack a frame burst full of them on a constant basis.

Remember, these limits only occur in half-duplex mode due to the round-trip timing requirements. Carrier extension is not needed in full-duplex mode, since full-duplex mode does not use CSMA/CD and is unconcerned about round-trip timing. A full-duplex Gigabit Ethernet system can operate at the full frame rate for all frame sizes, or ten times faster than a full-duplex Fast Ethernet system. Gigabit Ethernet performs quite well, particularly since the vendors of Gigabit Ethernet equipment support only the full-duplex mode of operation.

Collision Domain

A useful concept to keep in mind while working with Ethernet is the notion of *collision domain.* This term refers to a single half-duplex Ethernet system whose elements (cables, repeaters, station interfaces and other network hardware) are all part of the same signal timing domain. In a single collision domain, if two or more devices transmit at the same time a collision will occur. A collision domain may encompass several segments, as long as they are linked together with repeaters, as shown in Figure 3-4.

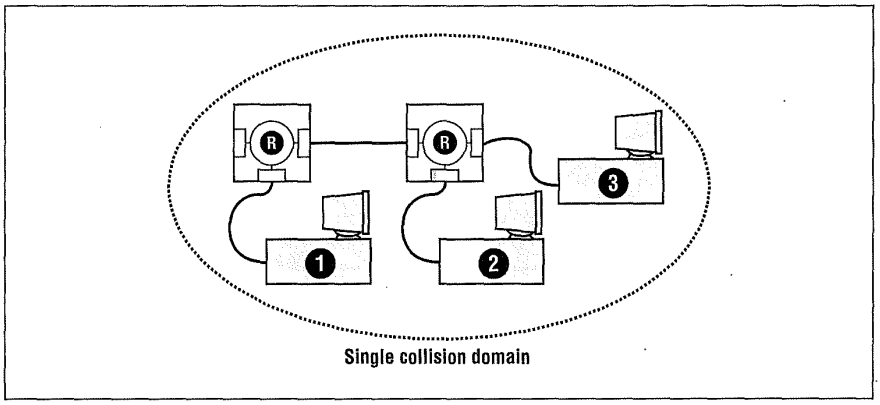


Figure 3-4. Ethernet collision domain

A repeater is a signal-level device that enforces the collision domain on the segments to which it is connected. The repeater only concerns itself with individual Ethernet signals; it does not make any decisions based on the addresses of the frame. Instead, a repeater simply retransmits the signals that make up a frame.

Repeaters make sure that the repeated media segments are part of the same collision domain by enforcing any collisions seen on any segment attached to the

repeater. For example, a collision on segment A is enforced by the repeater sending a jam sequence onto segment B. As far as MAC protocol (including the collision detection scheme) is concerned, a repeater makes multiple network cable segments function like a single cable. Repeaters are described in more detail in Chapter 17, *Ethernet Repeater Hubs.*

On a given Ethernet composed of multiple segments connected with repeaters, all of the stations are involved in the same collision domain. The collision algorithm is limited to 1024 distinct backoff times. Therefore, the maximum number of stations allowed in the standard for a multi-segment LAN linked with repeaters is 1024. However, that doesn't limit your site to 1024 stations, because Ethernets can be connected together with packet switching devices such as switching hubs or routers.

As Figure 3-5 illustrates, the repeaters and computers are connected by means of a *switching hub.* These Ethernets are in separate collision domains since switching hubs do not forward collision signals from one segment to another. Switching hubs contain multiple Ethernet interfaces. They operate by receiving a frame on one Ethernet port, moving the data through the hub, and transmitting the data out another Ethernet port in a new frame.

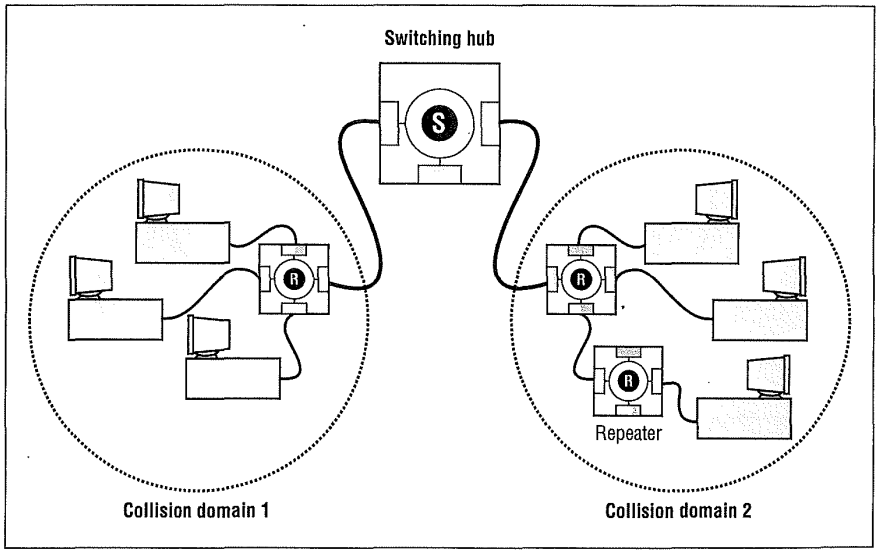


Figure 3-5. A switching hub is used to create separate collision domains

When you use a switching hub to link Ethernets, the linked Ethernets will each have their own separate collision domain. Unlike Ethernet repeaters, switching hubs do not enforce collisions onto their attached segments. Instead, individual

Ethernet systems attached to a switching hub function as separate network systems. Each Ethernet connected to a switch port can be as large as the standard allows. This means that you can use switches to link multiple Ethernets without being concerned about the sum of all stations attached to all of the Ethernets exceeding 1024 stations.

As long as switching hubs or routers are used to link the Ethernets, then the round-trip times and collision domains of those Ethernets are kept separate. This allows you to link a whole campus of Ethernet LANs without worrying about exceeding the guidelines for a single collision domain. You can use switching hubs to build large network systems by interconnecting individual Ethernet systems. The operating details of switching hubs are described in Chapter 18.

Ethernet Channel Capture

The Ethernet MAC protocol is a reliable, low overhead access control system that has proved its worth in millions of Ethernets. However, the MAC protocol is not perfect, and there are aspects of its operation that are not always optimal.

The best known example of this is an effect called *Ethernet channel capture.* Channel capture results in short-term unfairness, during which a station tends to consistently lose the competition for channel access. There must be one or more stations with a lot of data to send for channel capture to occur, causing them to contend for access the channel.

The sending station must also be able to continually transmit data at the maximum rate supported by the channel for short periods of time. Continuously sending data in this way sets up the collision and backoff conditions required for channel capture to occur. If the sending station does not have enough performance to continuously send data at the full rate supported by the channel, then the capture effect will not occur.

Operation of Channel Capture

An example of *channel capture* works as follows. If you look at all the stations on a channel when several active stations are simultaneously contending for access, you would expect to see collisions. Each station will possess a nonzero collision counter. As soon as one of the stations acquires the channel and delivers its frame, it clears its collision counter and starts over with a new frame transmission. The rest of the stations trying to transmit will still have nonzero collision counters.

If the *winning* station immediately returns to the channel contention with a collision counter of zero, it has an advantage over the other stations which have higher collision counters. The stations with higher collision counters will tend to choose

longer backoff times before retrying their frame transmissions. The station that wins will return to the fray with a zero collision counter, and therefore tend to continue winning. This station will effectively *capture* the channel for a brief period.

This can only occur if the winning station can rapidly and continually transmit data, which requires a high-performance station with a lot of data to send. This is not a common scenario, since most stations typically send data in short bursts. This helps explain why channel capture was first noticed when artificially high network loads were created using performance test software. Another place channel capture may be seen is when a file server is generating large bulk data flows doing backups while various user machines are trying to access the same channel.

Let's look at a worst-case example of channel capture. Consider the case of two high-performance workstations which both have a significant amount of data to send. Both computers also have Ethernet interface cards and computer busses that permit them to send frames as fast as the Ethernet channel allows. On their first transmission they collide and choose a backoff of zero or one. Let's say Station A chooses backoff 0 and Station B chooses 1. In this case, A would transmit the frame while B waits for its 1 slot time. At the end of A's frame transmission, both A and B are ready to transmit again. They both wait for the interframe gap period, then transmit, collide, and backoff. This is A's first collision for this frame transmission attempt, so A will choose a backoff of 0 or 1. Assuming this is B's second collision, then B will choose a backoff integer from between 0 and 3.

There is a five in eight chance that A will pick a lower number than B; a two in eight chance that they choose the same number and collide again; and a one in eight chance that B will choose a lower number than A. Therefore, A will most likely win by picking a lower backoff time and transmit its frame. If they pick the same number and collide again, the odds get even worse for B.

Since we're assuming that these two stations have a lot of data to send, the process repeats again. Only this time it is still A's first attempt to transmit a new frame and poor station B is on attempt number three for its frame transmission. Each time station A succeeds, station B's disadvantage increases. Once A has transmitted three or four frames, then A will pretty much be able to transmit at will. B will continue losing the channel contentions until B's transmission attempts counter reaches 16 and station B discards the frame and starts over. At this point A and B are back on even terms, and the contention is fair again.

Long-Term Fairness

If you look at the channel arbitration during the 16 retries that B makes trying to send a frame, then the channel appears to be unfair. But over a period of several

minutes both A and B will get about equal shares of the channel, since sometimes B will be the winner that gets to send a burst of packets. For most stations it does not appear that channel capture often results in a frame discard. Instead, the typical network application will tend to have less than 16 frames worth of data to send at any given time, and the channel capture will be of shorter duration.

Stations with slower network interfaces that cannot sustain a constant train of back-to-back frame transmissions will tend to break up the capture effect. As soon as a station ever pauses in its transmission attempts, then another station will be able to access the channel. Further, networks with high-performance computers are often segmented into smaller pieces using switching hubs, and the smaller populations on these segmented network channels are less likely to get involved in channel capture.

To really see channel capture at work may take an artificially high load, which is why channel capture may be seen with applications that test network throughput using a constant stream of data being sent between network test programs. For example, if channel capture occurs while measuring network performance using IP software, you could try setting the size of the "window" in the IP network software down to something around 8 Kbytes. This has the effect of reducing the total number of packets that will be sent at any given time, while not making any significant difference in throughput over a 10 Mbps channel. This, in turn, breaks up the capture effect and allows the network test program to provide the high throughput results that everyone was expecting.

A Fix for Channel Capture?

Over the years, channel capture has been studied intensely because of the potential bottlenecks it could create. Subsequently, a mechanism called Binary Logarithmic Arbitration Method (BLAM) was developed as a way to avoid channel capture. The BLAM mechanism eliminates the capture effect by changing the backoff rules to make access to the channel more fair. The result is to smooth the flow of packets on a busy network. The BLAM algorithm was designed to be backward compatible with existing Ethernet interfaces, so that mixed networks with standard Ethernet and BLAM-equipped interfaces could interoperate. A complete description of Ethernet channel capture and the BLAM algorithm can be found in a paper by Mart L. Molle, titled *A New Binary Logarithmic Arbitration Method for Ethernet.**

An IEEE project called 802.3w was launched to study the implementation of BLAM and to decide whether to add BLAM to the official Ethernet standard. For a variety

* This paper is Technical Report CSRI-298 of the Computer Systems Research Institute, University of Toronto, Toronto, Canada, M5S 1A1. The paper is available online in PostScript format via anonymous FTP from *ftp://ftp.cs.toronto.edu/csri-technical-reports/298/* in two files: *paper.ps* and *figures.ps.*

of reasons it was decided not to standardize BLAM. For one thing, vendors were nervous about deploying a new Ethernet operational mode into the field. Despite simulations and lab tests of BLAM, the new algorithm had not been deployed extensively. What if unforeseen complications occurred? There are many millions of Ethernet nodes already in use, which makes vendors understandably conservative when it comes to changing how Ethernet works.

Meanwhile, reports from the field indicated that most customers were not encountering the capture effect, so it appeared that vendors would be going to significant effort and risk to solve a relatively rare problem. Many networks these days are segmented with switching hubs, thereby limiting the number of machines that are contending for access on any given channel and making it harder for channel capture to occur. It's also the case that higher speed Ethernet makes it more difficult for channel capture to occur. A given application and set of stations that can cause channel capture on a 10 Mbps network has to work ten times harder to capture the channel on a 100 Mbps network.

For that matter, high-performance computers and server systems in today's networks are typically connected directly to an Ethernet switch, with full-duplex mode of operation enabled on the link. Full-duplex mode does not use the Ethernet MAC protocol, which means the capture effect can never occur on these links. In the end, the shrinking benefits of adding BLAM appeared to be outweighed by the risks, and the BLAM spec was shelved.

High-level Protocols and the Ethernet Frame

A wide range of computers can use the same Ethernet system, with each computer "speaking" several different high-level network protocols. The process of identifying which high-level network protocol suite is being carried in the data field of the frame is called *multiplexing*. Multiplexing simply means that multiple sources of information can be placed onto a single system. In this case, multiple high-level protocols can be carried over the same Ethernet system.

Multiplexing Data in Frames

The original system of multiplexing for Ethernet is based on using the type field in the Ethernet frame. For example, the high-level protocol software can create a packet of IP data, and then hand the packet to the software on the computer that understands how to create Ethernet frames with type fields. The framing software inserts a hexadecimal value in the type field of the frame that corresponds to the type of high-level protocol being carried by the frame, and then hands the data to the interface driving software.

Best Effort Delivery

This is a good place to point out that the Ethernet MAC protocol does not provide a guaranteed data delivery service. Like most other LAN systems, Ethernet does not provide strict guarantees for the reception of all data. Instead, the Ethernet MAC protocol makes a "best effort" to deliver the frame without errors. If the channel becomes congested and the collision retry limit is reached, or if bit errors occur during transmission, then a frame may be dropped. Ethernet was designed this way to keep the basic frame transmission system as simple and inexpensive as possible, by avoiding the complexities of establishing guaranteed reception mechanisms at the link layer.

It is assumed that higher layers of network operation, such as TCP/IP, will provide the mechanisms needed to establish and maintain reliable data connections when required. Despite all this, the vast majority of Ethernets operate with very few bit errors or dropped frames. The physical signaling system is designed to provide a very low bit error rate. As long as the channel is not continually overloaded, the odds that the frame will make it to its destination without problems are quite high.

The Ethernet interface driver deals with the details of interacting with the Ethernet interface to send the frame over the Ethernet channel. When carrying packets from IP, the type field will be assigned the hexadecimal value 0x0800. The receiving station then uses the value in the type field to demultiplex the received frame.

In Figure 3-6, two computer systems, Michelson and Morley, are sending data to one another over an Ethernet. Each system supports both TCP/IP and AppleTalk. The TCP/IP system uses the type field. AppleTalk, on the other hand, uses a method of identifying frame data based on the LLC standard, which is described later in this chapter.

In the figure, Michelson wants to send data to Morley via a TCP/IP-based application (such as an email application or some automatic network software that the OS runs). The application hands the data to the TCP/IP software. The TCP/IP software creates a high-level protocol packet to transport the data, and hands the packet to the interface driver software for the particular network technology that Michelson is connected to: in this case, an Ethernet system. The interface driver software creates an Ethernet frame, containing the TCP/IP packet in the data field of the frame.

At this point we'll ignore the issue of how to find Morley's Ethernet address, and just assume that this has been accomplished. (The details of the address discovery process using the address resolution protocol (ARP) are described in Chapter 2.) The frame is then sent over the Ethernet, where it is received by the physical

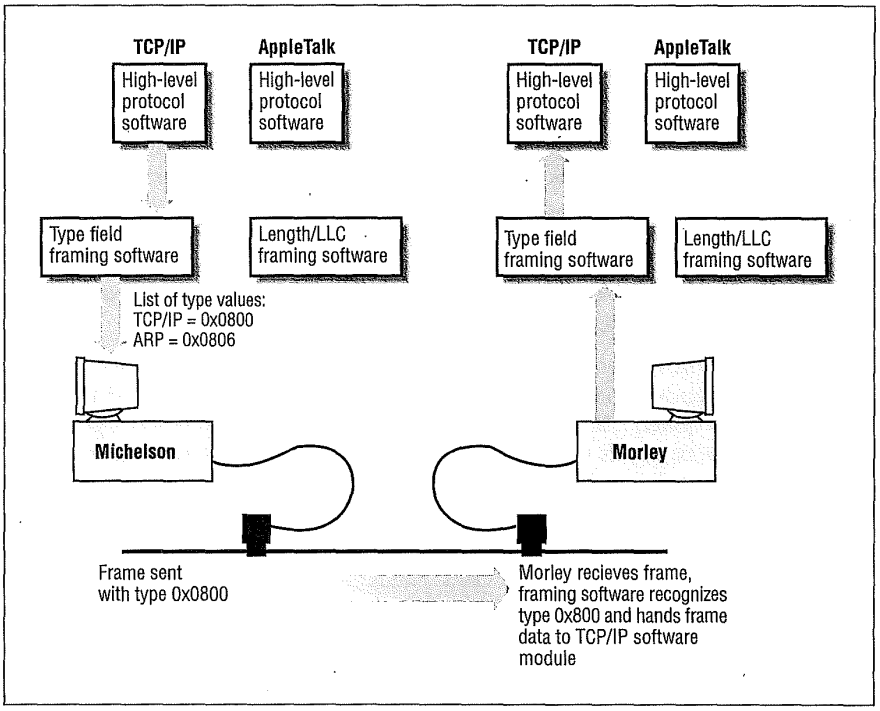


Figure 3-6. Multiplexing high-level protocols

network interface on Morley. The received frame is then handed to software that recognizes the type field, and hands the high-level protocol data carried by the frame to the appropriate high-level protocol software (such as TCP/IP). This is known as *demultiplexing.*

Each layer of the network system is substantially independent of the other layers. This independence is called *encapsulation.* By encapsulating the task performed by each layer, the complex system of network software can be broken down into more manageable chunks. By providing standardized operating system interfaces to the network programmers, the complexity of each network layer is effectively hidden from view.

The programmer is free to write software that hands the completed high-level protocol packet to the appropriate computer system software interface. The details of placing the protocol packet into the data field of an Ethernet frame are automatically dealt with. In this way, an IP-based application, and the IP software itself, can function without major changes regardless of which physical network system the computer happens to be attached to.

Things are made somewhat more complex because of the presence of two methods of identifying data in a frame: one using a type field to identify data, and one using the IEEE 802.2 Logical Link Control (LLC) standard. As you might expect, the mechanism used to identify data in the frame must be agreed upon by all the computers running the network software in question.

The most widely used high-level protocol today is TCP/IP, which uses the type field in the Ethernet frame. IP network software was originally developed on computers using the DIX Ethernet standard, which specified the use of a type field. IP network software continues to use the type field to the present day. By sticking with the type field, the IP software can continue to interoperate with other IP systems already in the field. Newer network protocols developed since the creation of the IEEE 802.2 LLC standard (e.g., AppleTalk), use the length field and LLC mechanisms for multiplexing and demultiplexing frame data.

Multiplexing Data with LLC

Since some network protocols use the LLC standard, you may encounter these frames when looking at the output of a network analyzer. To see how they work, we'll look next at the operation of a frame that uses the length/type field as a length field along with the LLC protocol.

As we've seen, the value of the identifier in the length/type field determines which way the field is being used. When used as a length field, the task of identifying the type of high-level protocol being carried in the frame is moved to the 802.2 LLC fields carried in the first few bytes of the data field. Let's look at the LLC fields in a little more detail (see Figure 3-7, below).

Figure 3-7 shows an IEEE 802.2 LLC protocol data unit, or PDU. The LLC PDU contains a Destination Service Access Point (DSAP), which identifies the high-level protocol that the data in the frame is intended for, much like the type field does. After a Source Service Access Point (SSAP) and some control data, the actual user data (the data that makes up the high-level protocol packet) follows the LLC fields.

When you are using the 802.2 LLC fields, multiplexing and demultiplexing work in the same way that they do for a frame with a type field. The difference is that the identification of the type of high-level protocol data is shifted to the SSAP, which is located in the LLC PDU. The whole LLC PDU fits inside the first few bytes of the data field of the Ethernet frame. In frames carrying LLC fields, the actual amount of high-level protocol data that can be carried is a few bytes less than in frames that use a type field.

You may be wondering why the IEEE went to all the trouble of defining the 802.2 LLC protocol to provide multiplexing when the type field seems to be able to do the job just as well. The reason is that the IEEE 802 committee was created to

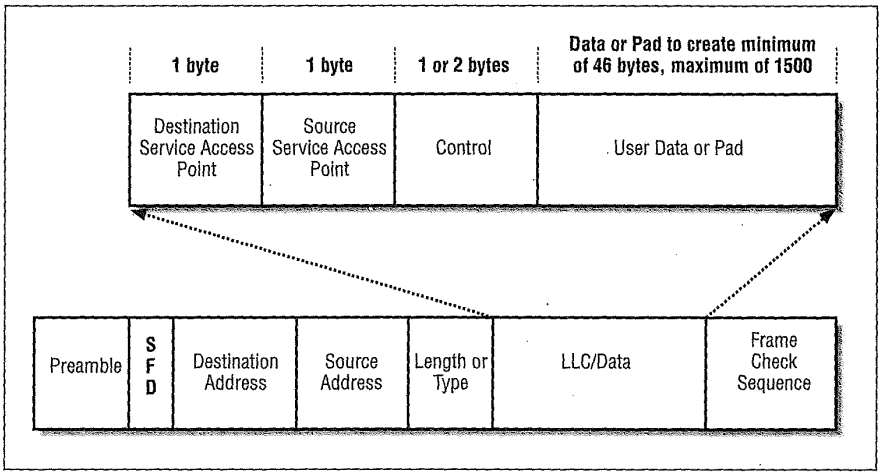


Figure 3-7. LLC PDU carried in an Ethernet frame

standardize a whole set of LAN technologies, and not just the 802.3 Ethernet system. To do that, they needed something that would work no matter which LAN technology was in use.

Since there was no guarantee that all LAN frames would have a type field, the IEEE 802 committee moved the task of identifying the type of data being carried by the frame into a LLC protocol. The LLC protocol defines a set of fields that are placed in the first few bytes of the data field. All LAN systems have a data field, which makes it easy enough to write network protocol software that can look at the first few bytes of data in the data field. The first few bytes of data are then interpreted in terms of the LLC specifications.

The interpretation of LLC fields is therefore identical no matter which LAN system is used. High-level network protocol implementations that were developed since the creation of the 802.2 standard in the 1980s typically use the LLC mechanisms. Therefore, if your site has a mix of computers running IP and, say, AppleTalk protocols, you can expect to see both frames with type fields, and frames with length fields and LLC data, traveling over your Ethernet.

LLC Sub-Network Access Protocol

Just to make things more interesting, the 802.2 LLC protocol can also be used to carry type identifiers. In other words, when you send a frame on a non-Ethernet LAN technology that does not provide a type field in its frame, there's a way to use the LLC fields to provide a type identifier. The rationale for this approach comes from the fact that the LLC fields are not large. Given that limitation, the

IEEE didn't want to use up the limited number of bits in the LLC fields to provide identifiers for all of the older high-level protocol types. Instead, a method was created to preserve the existing set of high-level protocol type identifiers, and to reuse them in the IEEE LLC system.

This approach provides yet another set of bytes in the data field of the frame, known as LLC Sub-Network Access Protocol (SNAP) encapsulation. In SNAP, the contents of the LLC fields of the frame are used to identify another set of bits in the data field that are organized according to the SNAP specification. The SNAP fields are used to carry the older protocol type identifiers. The standard for the use of SNAP encapsulation via IP is documented in RFC 1042. Access to RFC documents is described in Appendix A, *Resources.*

If you're writing network protocol software, then SNAP encapsulation is a handy way to continue using the same high-level protocol type identifiers when sending frames over other LAN systems. In the Ethernet system itself, of course, you can simply use a frame with the type field, and you don't need to concern yourself with any of this. However, you will probably encounter SNAP encapsulation if you deal with multiple LAN systems at this level of detail.

As a network user, you don't need to lose sleep over which frame format your computers may be using. The choice of frame format is typically built into your networking software, in which case there's nothing you need to do about it.

In this chapter:

- *Attachment Unit Interface*
- *Medium-Independent Interface*
- *Gigabit Medium-Independent Interface*
- *Ethernet Signal Encoding*
- *Ethernet Network Interface Card*

6

Ethernet Media Fundamentals

To send Ethernet signals from one station to another, stations are connected to one another with a media system based on a set of standard components. Some of these are hardware components specific to each media cabling system, such as the media cables and connectors. These media-specific components are described in the individual media chapters and cabling chapters that follow. Other components, such as Ethernet interfaces, are common to all media systems. Reading this chapter will provide the background you need to understand the components that connect to each of the Ethernet systems: 10-, 100-, or 1000 Mbps.

As the Ethernet system has evolved, it has developed a set of *medium-independent* attachments. Medium independence means that the Ethernet interface does not have to know anything about the media system. These attachments allow an Ethernet interface to be connected to any type of media system. With this system, multiple media systems can be developed without requiring any changes in the Ethernet interface controllers.

The first medium-independent attachment was developed for the 10 Mbps Ethernet system, and is called the *attachment unit interface* (AUI). The AUI supports the 10 Mbps media systems only. The next medium-independent attachment was developed as part of the Fast Ethernet standard, and is called the *medium-independent interface* (MII). The MII provides support for both 10 and 100 Mbps media segments. Finally, a *gigabit medium-independent interface* (GMII) was developed as part of the Gigabit Ethernet system. The GMII accommodates the increased speed of the Gigabit Ethernet system by providing a wider data path to the Ethernet interface. We'll look at all three of these medium-independent attachments, so that you can see how they are used to connect a station to an Ethernet.

This chapter also describes other elements of the Ethernet system that may be found in all media types. These include transceivers—both internal and external—

as well as transceiver cables and *network interface cards* (NICs). Finally, we will look at the block encoding schemes and signaling mechanisms used to send Ethernet signals over cables.

Attachment Unit Interface

The AUI was developed as part of the original 10 Mbps Ethernet system. Ethernet originated as a 10 Mbps system, operating over coaxial cable.* Later, new media systems based on twisted-pair and fiber optic link segments were invented for the 10 Mbps system. In the early 1990s, the 10BASE-T twisted-pair system became a popular method of connecting desktop computers, which lead to the widespread adoption of Ethernet as the networking system of choice.

The AUI makes it possible to connect an Ethernet interface to any one of the several 10 Mbps media systems while isolating the interface from any details of the specific media system in use. The development of a medium-independent attachment was actually a side effect of the design of the original thick coaxial cabling system. The thick coaxial system requires the use of external transceivers connected directly to the coax cable. Providing a connection between the Ethernet interface in the station and the external transceiver on the coaxial cable led to the development of the AUI. The AUI, in turn, made it possible to develop other cabling systems for 10 Mbps Ethernet without requiring any changes in the Ethernet electronics in the station.

Figure 6-1 illustrates how the AUI is used in the 10 Mbps system. This figure shows the complete set of components that can be used to implement a twisted-pair 10 Mbps Ethernet connection. Both internal and external transceiver connections are shown.

This set of components and their three-letter acronyms might seem like alphabet soup at first glance. However, it is useful to know these acronyms as they are found in vendor literature and are printed on Ethernet devices. Starting with the DTE, we will then discuss the AUI connector and AUI cable, and finish with the transceiver.

Data Terminal Equipment or Repeater Port

To begin with, there is the networked device, which can be a station or a repeater port. A station, more formally called *data terminal equipment* (DTE) in the

* Ethernet coaxial cable systems provide only a single 10 Mbps channel, and coaxial cable systems can be difficult to install and operate. For that reason, they are no longer the first choice for implementing a network that must support more than a few stations. The details of coaxial Ethernet media systems (10BASE5 and 10BASE2) are covered in Appendix B, *Thick and Thin Coaxial Media Systems.*

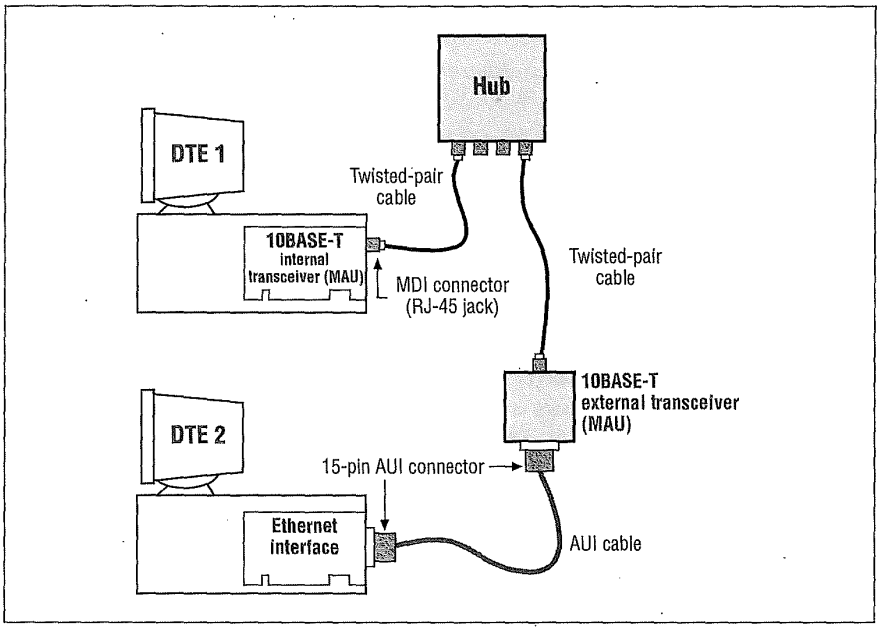


Figure 6-1. AUI connection for a 10 Mbps Ethernet system

standard, is a unique, addressable device on a network. A DTE is a device that serves as an originating or terminating point for data. For example, each Ethernet-equipped computer or port on a switching hub is a DTE, since each is equipped with an Ethernet interface. The Ethernet interface contains the electronics needed to perform the *media access control* (MAC) functions required to send and receive Ethernet frames over the Ethernet channel.

Ethernet ports on repeater hubs are *not* DTEs and do not use an Ethernet interface. A repeater port connects to an Ethernet media system using standard components, such as a transceiver. However, repeater ports operate at the individual bit level for Ethernet signals, moving the signals through the repeater so that they can travel from one segment to another. Therefore, repeater ports do not contain Ethernet interfaces since they do not operate at the level of Ethernet frames.

Attachment Unit Interface and Cable

The AUI is the medium-independent attachment that allows an Ethernet interface to be connected to one of several media systems. In Figure 6-1, DTE 2 has an AUI

connector and external transceiver that, in turn, is connected to a twisted-pair cable. The AUI connector on DTE 2 makes it possible for that station to be connected to any of several 10 Mbps Ethernet media systems, by using the appropriate external transceiver. Figure 6-1 also shows DTE 1, which has a built-in 10BASE-T transceiver. Since this station does not have a 15-pin AUI connector, it cannot be connected to any other media system; it connects only to a twisted-pair cable.

AUI Connector

The 15-pin AUI connector provides an external transceiver connection for a station. This connector provides power to the external transceiver, and provides a path for Ethernet signals to travel between the Ethernet interface and the media system. The AUI connector uses a slide latch mechanism to make an attachment between male and female 15-pin connectors. The slide latch mechanism is described in Appendix C, *AUI Equipment: Installation and Configuration.* Figure 6-2 lists the signals provided by the 15-pin AUI connector.

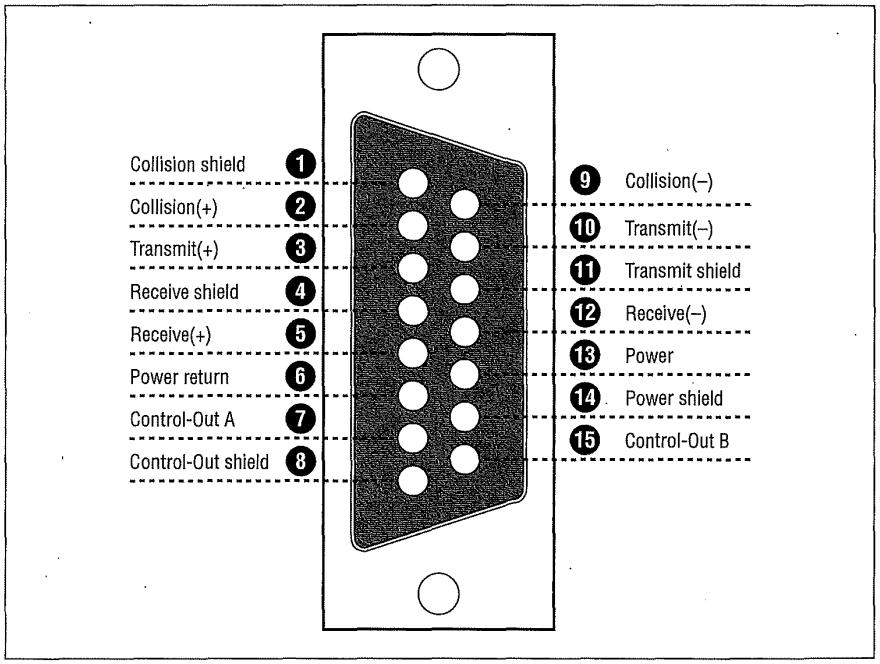


Figure 6-2. AUI connector signals

The signals sent by the transceiver and received by the Ethernet interface are low voltage differential signals. There are two wires for each signal, with one wire for the positive (+) and one wire for the negative (-) portions of the signal. The voltage level on these wires varies from +0.7 volts to –0.7 volts, providing a nominal total of 1.4 volts peak-to-peak for the entire signal.

The "Control-Out" signal was provided to send optional control signals from an Ethernet interface to the transceiver. This option was never implemented by any vendor and is not used.

AUI Transceiver Cable

The 10 Mbps transceiver cable, formally known as an AUI cable, is built like an electrical extension cord: there's a plug (*male connector*) on one end and a socket (*female connector*) on the other. The transceiver cable carries three data signals between a 10 Mbps Ethernet interface and the external transceiver:

- *Transmit data*, from the Ethernet interface to the transceiver.
- *Receive data*, from the transceiver to the interface.
- A *collision presence signal*, from the transceiver to the interface.

Each signal is sent over twisted-pair wires. Another pair of wires is used to carry 12-volt DC power from the Ethernet interface to the transceiver. The standard transceiver cable uses fairly heavy-duty stranded wire to provide good flexibility and low resistance.

The AUI transceiver cable, shown in Figure 6-3, is equipped with a 15-pin female connector on one end that is provided with a sliding latch; this is the end that is attached to the outboard transceiver. The other end of the transceiver cable has a 15-pin male connector, equipped with locking posts; this is the end that attaches to the Ethernet interface. Some 15-pin AUI connectors on Ethernet interfaces have been equipped with screw posts instead of the sliding latch fastener described in the standard, requiring a special transceiver cable with locking screws on one end instead of sliding latch posts.

The AUI transceiver cable described in the IEEE standard is relatively thick (approximately 1 cm or 0.4 inch diameter), and may be up to 50 meters (164 feet) long. There is no minimum length standard for a transceiver cable, and external transceivers are sold that are small enough to fit directly onto the 15-pin AUI connector of the Ethernet interface, dispensing with the need for a transceiver cable.

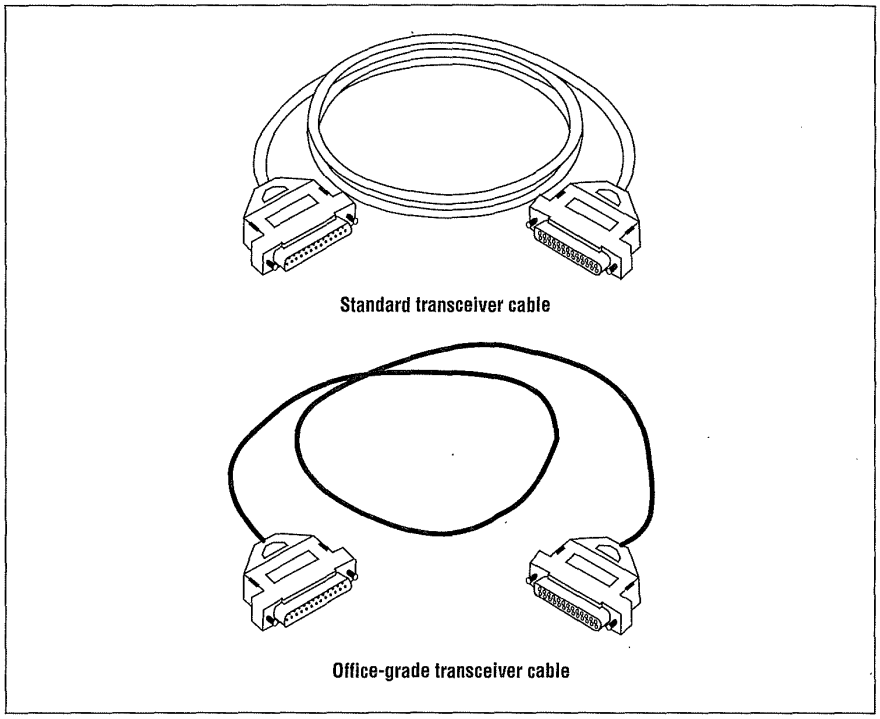


Figure 6-3. Standard and office-grade AUI transceiver cables

"Office-grade" transceiver cables (shown at the bottom of Figure 6-3), are thinner and more flexible than the standard cable. The thinner wires used in office grade transceiver cables also have higher signal loss than standard cables, which limits the length of office grade cables. The maximum allowable length for an office grade transceiver cable, which has four times the amount of signal attenuation as standard cables, is 12.5 meters (41 feet).

Theoretically, you could connect several transceiver cables together to make up a single longer cable. However, this may not be a good idea, since the sliding latch connectors may not hold the cable ends together very well, and you could end up with intermittent connections.

Medium Attachment Unit

The next component shown in the DTE 2 connection in Figure 6-1 is the *medium attachment unit* (MAU), more commonly known as a *transceiver*. The transceiver gets its name because it *trans*mits and re*ceives* signals on the physical medium. The AUI transceiver is the link between the different types of electrical signaling

used over the media systems, and the signals that are sent through the AUI interface to the Ethernet interface in the station. Each 10 Mbps media system has a specific transceiver designed to perform the type of electrical signaling used for that medium. There are coaxial, twisted-pair, and fiber optic transceivers, each equipped with the components it takes to send and receive signals over that particular medium.

The external AUI transceiver is a small box, typically a few inches on each side. There's no specified shape; some are long and thin, and some are almost square. The transceiver electronics typically receive their power over the transceiver cable from the Ethernet interface in the station. According to the standard, an AUI transceiver may draw up to 500 milliamps ($1/2$ amp) of current.

The transceiver transmits signals from the Ethernet interface onto the media, and receives signals from the media which it sends to the Ethernet interface. The signals that transceivers send vary according to the type of media in use. On the other hand, signals that travel between the transceiver and the Ethernet-equipped device over the AUI interface are the same, no matter which media type is in use. That's why you can make a connection between any 10 Mbps media system and a 15-pin connector on an Ethernet device. The signaling over the 15-pin interface is the same for all transceivers; only the media signals are different.

AUI transceiver jabber protection

The jabber protection function senses when a broken Ethernet device has gone berserk and is continuously transmitting a signal—a condition known as *jabbering*. Jabbering causes a continual carrier sense on the channel, jamming the channel and preventing other stations from being able to use the network. If that happens, the jabber protection circuit enables a *jabber latch*, which will shut off the signal to the channel.

The AUI transceiver specification allows two methods of resetting the jabber latch: power-cycling or by automatically restoring operation one-half second after the jabbering transmission ceases. In some very old AUI transceivers, the jabber latch would not reset until the transceiver was power-cycled, which required the network administrator to disconnect and reconnect the transceiver cable to get the transceiver to work again. Needless to say, this approach was not very popular with network administrators. Modern AUI transceivers are built using a single chip design that automatically comes out of jabber latch mode once an overlong transmission has ceased.

The SQE Test signal

The earliest Ethernet standard, DIX V1.0, did not include a signal for testing the operation of the collision detection system. However, in the DIX V2.0 specifications,

the AUI transceiver was provided with a new signal, called the *collision presence test* (CPT). The name for the collision signal changed to *signal quality error* (SQE) in the IEEE 802.3 standard, and the CPT signal was also changed to SQE Test. The purpose of the SQE Test signal is to test the collision detection electronics of the transceiver, and to let the Ethernet interface know that the collision detection circuits and signal paths are working correctly.

When you install an external AUI transceiver on your Ethernet system it is extremely important to correctly configure the SQE Test signal. The SQE Test signal *must be disabled* if the transceiver is attached to a repeater hub. For all other devices that may be attached to an external 10 Mbps transceiver, the standard recommends that the SQE Test signal be enabled.

You may find that some vendors do not label things correctly, which can lead to some confusion. For example, you may find that the switch on the transceiver for enabling the SQE Test signal will be labeled "SQE" instead of "SQE Test." Since "SQE" is the name of the actual collision signal, the last thing you'd want to do is disable the collision detection signal in Ethernet. Nonetheless, this confusion of terms is very widespread. The operation and configuration of the SQE Test signal is described in detail in Appendix C.

Medium-Dependent Interface

The actual connection to the network medium (e.g., twisted-pair cable) is made by way of a component that the standard calls the *medium-dependent interface*, or MDI. In the real world, this is a piece of hardware used for making a direct physical connection to the network cable.

In Figure 6-1, the MDI is an eight-pin connector, also referred to as an RJ-45–style jack. The MDI is actually a part of the transceiver, and provides the transceiver with a direct physical and electrical connection to the twisted-pair wires used to carry network signals in the 10 Mbps twisted-pair media system.

In the case of thick coaxial Ethernet, the most commonly used MDI is a type of coaxial cable clamp that is installed directly onto the coaxial cable. For fiber optic Ethernet, the MDI is a fiber optic connector.

Medium-Independent Interface

The invention of the 100 Mbps Fast Ethernet system was also the occasion for the development of a new attachment interface, called simply the *medium-independent*

interface (MII). The MII can support operation at both 10 and 100 Mbps. Figure 6-4 shows the same two stations as Figure 6-1, but this time an MII is being used.

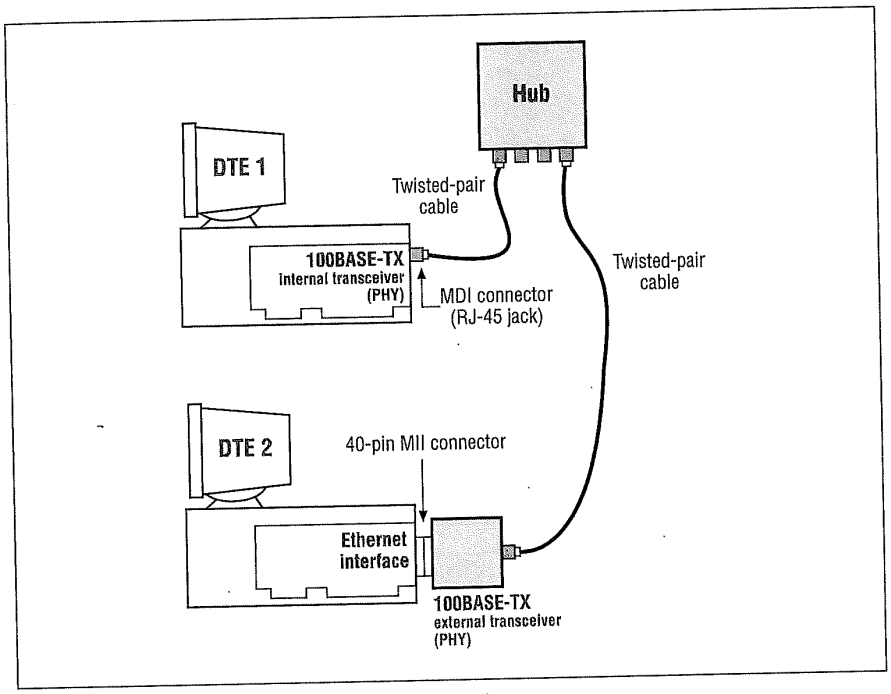


Figure 6-4. MII connection for a 100 Mbps Fast Ethernet system

The other major difference between this diagram and the one shown in Figure 6-1 is that a transceiver with an MII is called a *physical layer device* (PHY) instead of a MAU. In essence, the MII is an updated and improved version of the original 10 Mbps–only AUI. The MII can be embedded inside equipment, as in the twisted-pair interface shown on DTE 1. In this case, the transceiver and MII are inside the computer. All the user sees is the twisted-pair connector, which connects the twisted-pair media to the internal transceiver.

An Ethernet device can also be provided with a 40-pin MII connector, which allows connections to external transceivers, as shown attached to DTE 2 in Figure 6-4. The external transceiver provides maximum flexibility, since you can provide either a twisted-pair or fiber optic transceiver. This allows a connection to either twisted-pair or fiber optic media types operating at 10 or 100 Mbps speeds.

The MII is designed to make the signaling differences among the various media segments transparent to the Ethernet electronics inside the networked device. The

MII does this by converting the signals received from the various media segments by the transceiver (PHY) into standardized digital format signals. The digital signals are then provided to the Ethernet electronics in the networked device over a 4-bit wide data path. The same standard digital signals are provided to the Ethernet interface no matter what kind of media signaling is used.

MII Connector

The 40-pin MII connector and optional MII cable provide a path for the transmission of signals between an MII interface in the station and an external transceiver. The vast majority of MII transceivers are designed for direct connection to the MII connector on the networked device, and do not use an MII cable.

Figure 6-5 shows two MII transceivers, one equipped with an MII cable (at top), and the other (below) equipped with jack screws for direct connection to the mating screw locks on the DTE. The jack screws replace the much-maligned slide latch mechanism used for the 15-pin AUI in the original 10 Mbps Ethernet system. If an optional MII cable is used, the end of the MII cable will be equipped with a 40-pin connector and a pair of jack screws that fasten into the mating screw locks on the networked device.

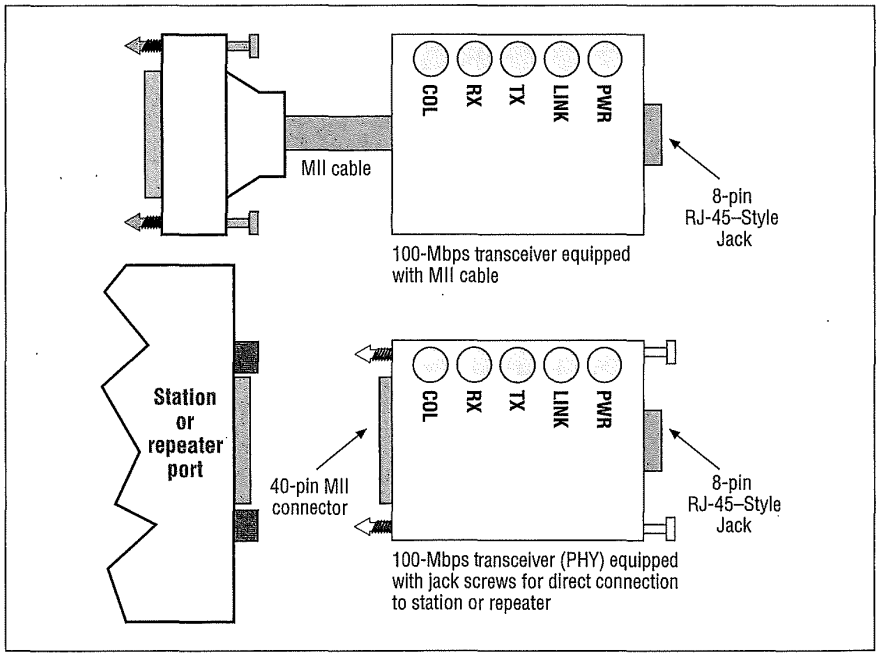


Figure 6-5. MII connector and transceiver

MII connector signals

The signals provided on the MII connector are different from the ones found on the 15-pin AUI connector in the 10 Mbps system.

The 40-pin MII connector is small and the pins are densely packed, so care should be taken to avoid damaging the pins when connecting and disconnecting network components. Note that the MII pins can be easily bent and that the +5 volt pins on the lower row are right next to ground pins.

If the +5 volt pins or ground pins bend and touch against one another during installation, it is possible to blow a fuse in the network equipment, which will cause the MII port to stop working. The prudent network manager may wish to power off the equipment while connecting or disconnecting MIIs, just to be safe.

Figure 6-6 shows the 40 pins of the MII connector, with the signals carried by the pins indicated. The MII defines a 4-bit wide data path for transmit and receive data that is clocked at 25 MHz to provide a 100 Mbps transfer speed, or 2.5 MHz for 10 Mbps operation. According to the standard, the electronics attached to each MII connector (male and female) should be designed to withstand connector insertion and removal while the power is on.

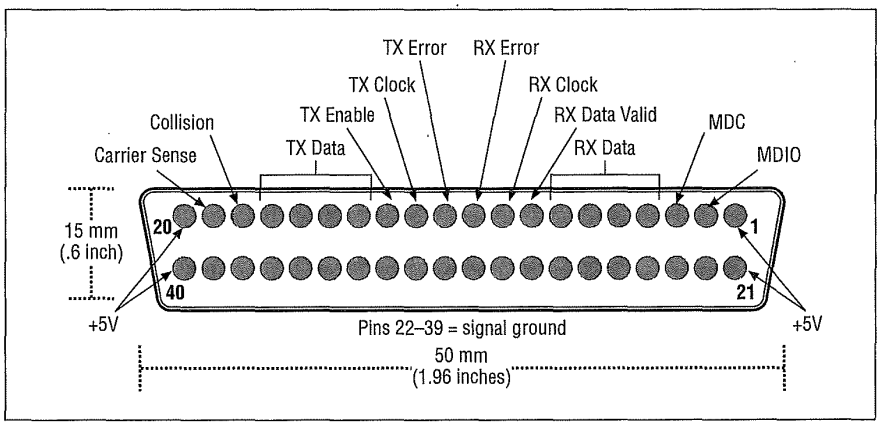


Figure 6-6. MII signals

The MII provides for a set of control signals that make it possible for the Ethernet interface in the networked device to interact with the external transceiver to set and detect various modes of operation. This management interface can be used to

place the transceiver into loopback mode for testing, to enable full-duplex operation if the transceiver supports it, to select the transceiver speed if the transceiver supports dual speed mode, and so on. The MII signals are as follows:

+5 volts

Pins 1, 20, 21, and 40 are used to carry +5 volts at a maximum current of 750 milliamps, or 3/4 amp.

Signal ground

Pins 22 through 39 carry signal ground wires.

Management Data I/O

Pin 2 carries the Management Data Input/Output signal, a bi-directional signal used to carry serial data representing control and status information between the transceiver and the DTE. The management interface provides various functions, including resetting the transceiver, setting the transceiver into full-duplex mode, and enabling a test of the electronics and signal paths involved in the collision signal.

Management Data Clock

Pin 3 carries the Management Data Clock, which is used as a timing reference for serial data sent on the management data interface.

RX Data

Pins 4, 5, 6, and 7 provide the 4-bit Receive Data path from the transceiver to the DTE.*

RX Data Valid

Pin 8 carries the Receive Data Valid signal, which is generated by the transceiver while receiving a valid frame.

RX Clock

Pin 9 carries the Receive Clock which runs at 25 MHz in 100 Mbps Fast Ethernet systems and at 2.5 MHz in 10 Mbps systems, to provide a timing reference for receive signals.

RX Error

Pin 10 carries this signal, which is sent by the transceiver upon detection of received errors.

TX Error

Pin 11 carries this signal, which can be used by a repeater to force the propagation of received errors. This signal may be used by a repeater under certain circumstances and is never used by a station.

* A 4-bit chunk of data is also called a *nibble*, to distinguish it from an 8-bit byte.

TX Clock

Pin 12 carries the Transmit Clock, which runs continuously at a frequency of 25 MHz for 100 Mbps Fast Ethernet systems and at 2.5 MHz for 10 Mbps systems. The purpose of this signal is to provide a timing reference for transmit signals.

TX Enable

Pin 13 carries the Transmit Enable signal from the DTE to the transceiver to signal the transceiver that transmit data is being sent.

TX Data

Pins 14, 15, 16, and 17 provide the 4-bit wide transmit data path from the DTE to the transceiver.

Collision

Pin 18 carries this signal from the transceiver. It indicates a collision detected on the network segment. If a transceiver is in full-duplex mode, then this signal is undefined by the standard and the collision light on the transceiver may glow steadily or erratically when full-duplex mode is enabled. See the instructions that came with the transceiver you are using for details.

Carrier Sense

Pin 19 carries this signal, which indicates activity on the network segment from the transceiver to the DTE.

MII Transceiver and Cable

The PHY shown in Figure 6-4 performs approximately the same function as a transceiver in the 10 Mbps Ethernet system. However, unlike the original 10 Mbps MAU, the PHY, or transceiver, also performs the media system signal encoding and decoding. In addition, an MII transceiver can be automatically configured to operate in full- or half-duplex mode, and can operate at either 10 or 100 Mbps.

The transceiver may be a set of integrated circuits inside the Ethernet port of a network device and therefore invisible to the user, or it may be a small box like the external transceiver used in 10 Mbps Ethernet. An external MII transceiver is equipped with a 40-pin MII plug designed for direct connection to the 40-pin MII jack on the networked device, as shown in Figure 6-5. This connection may include a short MII transceiver cable, although in practice these are not easily found.

According to the standard, an MII cable consists of 20 twisted pairs with a total of 40 wires. The twisted-pair cable also has a 40-pin plug on one end equipped with male jackscrews that connect to mating female screw locks. The cable can be a maximum of 0.5 meters in length (approximately 19.6 inches). However, the vast

majority of external transceivers attach directly to the MII connector on the device with no intervening cable.

MII jabber protection

An MII transceiver operated at 10 Mbps has a jabber protection function, which provides a jabber latch similar to the one in the AUI transceiver described earlier in this chapter. In the 100 Mbps Fast Ethernet system, the jabber protection feature was moved to the Fast Ethernet repeater ports. This change was made possible because all Fast Ethernet segments are link segments and must be connected to a repeater hub for communication with other stations. Moving the circuitry for jabber protection to the repeater hub offloads that requirement from the transceiver and provides the same level of protection for the network channel. Therefore, transceivers operating at 100 Mbps Fast Ethernet do not provide the jabber latch function. Instead, each Fast Ethernet repeater port monitors the channel for long transmissions and shuts the port down if the carrier signal persists for anywhere from 40,000 to 75,000 bit times.

MII SQE Test

The SQE Test signal is provided on AUI-based equipment to test the integrity of the collision detect electronics and signal paths. However, there is no SQE Test signal provided in the MII. SQE Test can be removed from the MII since all media systems connected to an MII are link segments. Collisions are detected on link segments by the simultaneous occurrence of data on the receive and transmit data circuits. As such, the link monitor function in MII transceivers ensures that the receive data circuits are working correctly.

Additionally, the MII provides a loopback test of the collision detect signal paths from an external transceiver to the Ethernet device. Taken together, this provides a complete check of collision detect signal paths, making the SQE Test signal unnecessary for the MII.

Gigabit Medium-Independent Interface

The development of 1000 Mbps Gigabit Ethernet system led to the development of yet another attachment interface, called the *gigabit medium-independent interface* (GMII). Because of the higher speeds used in Gigabit Ethernet, an externally exposed interface is too difficult to engineer reliably. For this reason, the GMII does not support an exposed connector for attaching a transceiver or an outboard transceiver cable. Therefore, unlike the AUI or MII, the GMII only provides a standard way of interconnecting integrated circuits on circuit boards. At most, the GMII can be used as a motherboard-to-daughterboard interface inside a piece of equipment.

Since there is no exposed GMII, there is no way to attach an external transceiver to a Gigabit Ethernet system. If a station is equipped with a GMII, all the user will see is an MDI connector, such as the 8-pin RJ-45–style jack used for making the connection to a twisted-pair segment.

Like the MII, the GMII provides media independence by making the signaling differences among media segments transparent to the Ethernet electronics inside the networked device. The GMII converts the various media line signals received by the embedded Gigabit Ethernet transceiver (PHY) into standardized digital data signals. These signals are then provided to the Ethernet controller chip, which is responsible for creating Ethernet frames for transmission, and for assembling received data into Ethernet frames for reception. While the MII provides 4-bit data paths, the GMII uses a byte-wide interface, providing an 8-bit data path between the transceiver chip and the Ethernet controller chip inside the equipment. By transferring twice as much data for a given data interface clock rate, the GMII makes it easier for an Ethernet controller to handle gigabit data speeds.

The GMII is designed to support only 1000 Mbps operation, with 10 and 100 Mbps operation provided by the MII. Transceiver chips are available that simultaneously implement both the MII and GMII circuits on a given Ethernet port. This provides 10/100/1000 Mbps support over twisted-pair cabling with automatic configuration via the Auto-Negotiation protocol.

Gigabit Ten-Bit Interface

For Gigabit Ethernet devices that only support the 1000BASE-X media family, the media independence provided by the GMII is not required. The 1000BASE-X system is based on signaling and media systems originally developed for the ANSI Fibre Channel standard. If only 1000BASE-X support is needed, then another internal interface, called the *Ten-Bit Interface* (TBI), is provided by the standard. This interface is ten code bits wide, to accommodate the 8B/10B signal encoding used in the 1000BASE-X media system, described later in this chapter.

The TBI is used in commercial Fibre Channel serializer/deserializer (SERDES) components, and has been adopted by Ethernet chip manufacturers for use in 1000BASE-X systems. Like the GMII, the TBI is also hidden inside the equipment—only equipment designers need to deal with it. In practice, you will only see a fiber optic or copper Gigabit Ethernet port on the outside of the equipment. The exact details of which internal signaling interface is used and how the port is wired up inside the device are not important to the user of Gigabit Ethernet equipment.

GMII Transceiver

A Gigabit Ethernet transceiver (PHY) consists of one or more integrated circuit chips located inside the Gigabit Ethernet device. Since there is no exposed signaling interface for Gigabit Ethernet, there are also no external transceivers. However, some of the media signaling components used in a transceiver may be exposed to the user as part of the Gigabit Interface Converter (GBIC), which is described in the next section.

Like the MII transceiver, the Gigabit Ethernet PHY also performs media system signal encoding and decoding, and can be automatically set to operate in full- or half-duplex mode. Unlike the MII transceiver, the GMII transceiver is not capable of multi-speed operation, and is designed to only operate at 1000 Mbps.

Gigabit Interface Converter

As mentioned above, some of the media signaling components used to perform transceiver functions are exposed to the user as part of the GBIC. The GBIC was originally developed for use in the Fibre Channel system, and is referred to as a *serial transceiver module* in that system. Gigabit Ethernet vendors have adopted the Fibre Channel language and refer to GBICs as transceivers. Strictly speaking, the GBIC module only provides the transmitter and receiver components needed to send signals over Gigabit Ethernet, and does not function as a complete Gigabit Ethernet transceiver. Nonetheless, you will see the phrase "Gigabit Ethernet transceiver" used in reference to GBICs.

The GBIC is a small module that can be plugged into the Gigabit Ethernet port on a switch or an interface card and is hot-swappable. The GBIC and the components inside a GBIC port on the switch are static sensitive, so be sure to use a static grounding strap when handling or installing a GBIC.

The GBIC is a widely adopted way of providing the transmitter and receiver components needed to send signals over Gigabit Ethernet fiber optic media segments. Vendors also expect to provide 1000BASE-T GBICs as the twisted-pair Gigabit equipment evolves. Fiber optic GBIC modules make it possible for a given Gigabit Ethernet port to be equipped with either 1000BASE-SX or 1000BASE-LX fiber optic media signaling components as needed by the customer.

Figure 6-7 shows a GBIC designed for connection to a 1000BASE-LX Gigabit Ethernet fiber optic segment. There are two SC connectors on the front of the GBIC for connecting the fiber optic cables. The back of the GBIC has a 20-pin connector (called the "SC-20" connector in the GBIC standard), for connection to the electronics in the Gigabit Ethernet port. The GBIC snaps into place when fully inserted into a port on an Ethernet hub or interface card.

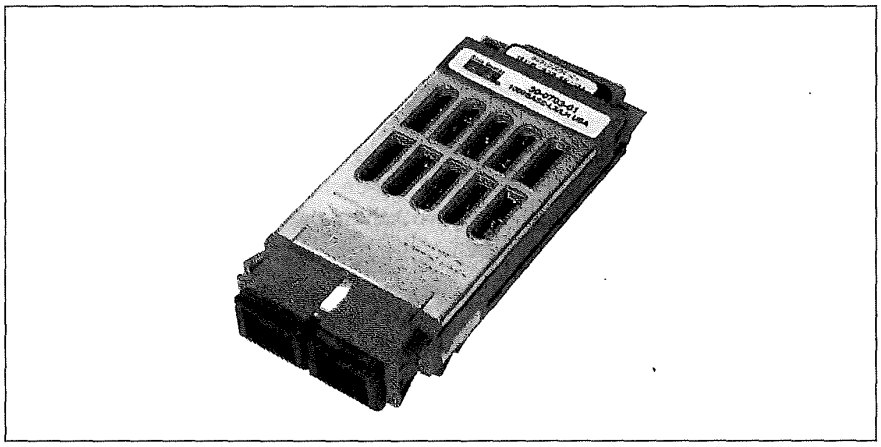


Figure 6-7. A Gigabit Interface Converter (GBIC)

Whether or not a Gigabit Ethernet switching hub or network interface port is equipped with a GBIC is entirely up to the vendor. Where cost or port density is a concern, the vendor may choose to use built-in SC or MT-RJ fiber optic connectors. On the other hand, if a vendor wants to provide maximum flexibility on a switching hub port, for example, then they may choose to use a GBIC.

GBICs are available from a number of sources. However, some major vendors of Gigabit Ethernet switches note that they will not support third-party suppliers. In that case, you must buy your GBICs from the same vendor that made the network equipment. In general, this is probably a good practice as the vendor can certify that the GBICs they sell for use in their equipment meet all of their requirements.

Ethernet Signal Encoding

A number of encoding schemes are used for sending Ethernet signals over the various types of media. Signal encoding is a means of combining both clocking and data information into a self-synchronizing stream of signals sent over a media system. Each media system presents a certain challenge to the engineer in terms of sending Ethernet signals that can make it from one end of the cable to another.

As higher speed Ethernet systems have evolved, more complex block encoding schemes have been used. All of these signaling systems have the same set of goals. One goal is to include sufficient clocking information along with the signals to ensure that the signal decoding circuitry can function correctly. Other goals are to ensure that the error rate is kept very low, and that the Ethernet signals have a very high probability of surviving their trip over the media system.

AUI Signal Encoding

Signals sent over all 10 Mbps media systems—these include the 10 Mbps coaxial, twisted-pair and fiber optic media systems—use a relatively simple encoding scheme called Manchester encoding.* Manchester encoding combines data and clock into *bit symbols*, which provide a clocking transition in the middle of each bit. As shown in Figure 6-8, each Manchester-encoded bit is sent over the network in a *bit period* which is split into two halves, the polarity of the second half always being the reverse of the first half.

The rules for Manchester encoding define a 0 as a signal that is high for the first half of the bit period and low for the second half. A 1 is defined as a signal that is low for the first half of the bit period, and high for the second half. Figure 6-8 shows a station sending the bit pattern 001..

One result of Manchester encoding is to provide a clock transition in each digital bit sent. This transition is used by the receiving station to synchronize itself with the incoming data. While Manchester encoding makes it easy for a receiver to synchronize with the incoming signal and to extract data from it, a drawback of the scheme is that the worst-case signaling rate is twice the data rate. In other words, a 10 Mbps stream of all ones or all zeroes results in a Manchester encoded signaling rate of 20 MHz on the cable.

Different *physical line signaling* methods are used to send the Manchester encoded signals over the media cable, depending on the media system involved. For example, Manchester encoded signals are sent over the original thick coax media system by transmitting an electrical current. Actually, a 10BASE5 transceiver sends two currents onto the coax: a steady DC offset current and a signaling current that changes in amplitude to represent ones and zeroes. The offset current provides a baseline around which the signals are sent. Should you look at thick coax signals with an oscilloscope, you might see something like the signals shown in Figure 6-8.

The line signaling current generates a signal voltage that ranges from 0 to –2 volts. Other line signaling schemes are used for 10 Mbps twisted-pair or fiber optic media systems. Manchester encoding is used for all 10 Mbps media systems, with the only difference being the line signaling system used to send the encoded bits over the cables. The specifics of each physical line signaling system are described in the individual media chapters.

* This system is named for its origin at Manchester University in England.

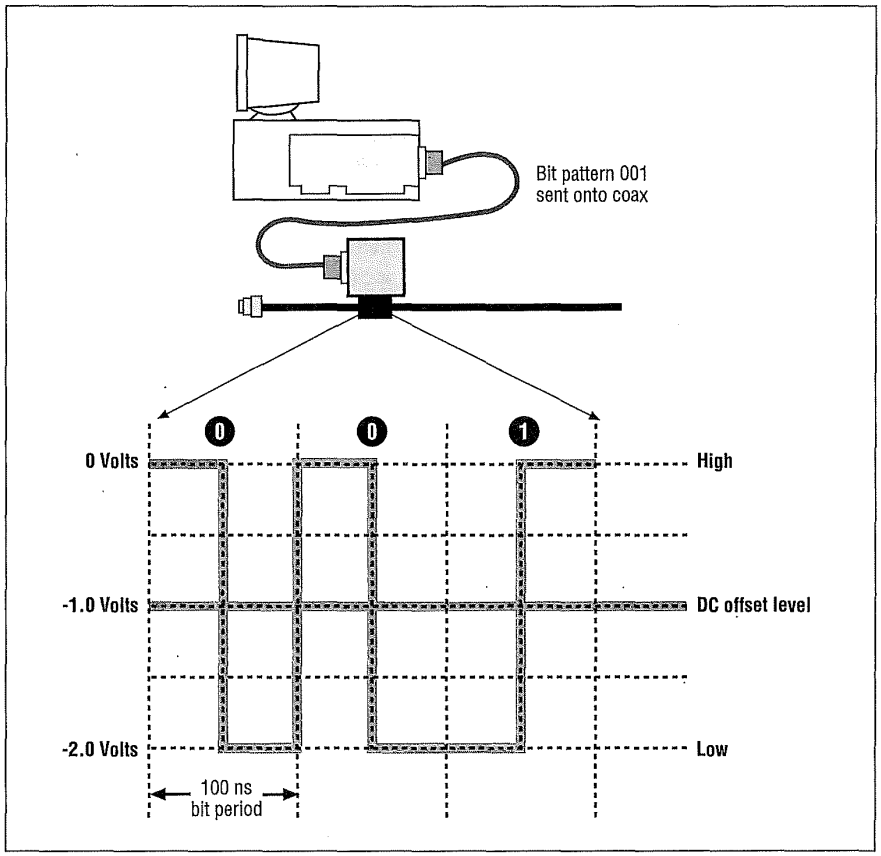


Figure 6-8. Manchester signals over 10BASE5

MII Signal Encoding

The MII transceivers are designed to deal with several signal encoding schemes. A twisted-pair MII transceiver can deal with both the Manchester encoding found on 10 Mbps systems, and the signal encoding used on 100 Mbps twisted-pair media systems. The MII transceiver turns these signals into standard 4-bit chunks of data, which are sent to the Ethernet controller.

The signal encoding used on Fast Ethernet systems is based on block encoding, which is more complex than Manchester encoding. Block encoding takes a group, or block, of data bits and encodes them into a larger set of code bits. The data stream is divided into a fixed number of bits per block, typically 4 or 8 bits. Each

data block is translated into a set of code bits, also called *code symbols*. For example, a 4-bit data block (16 possible bit patterns) may be translated into a 5-bit code symbol (32 possible symbols). The expanded set of code symbols is arbitrarily chosen, and is designed to help improve line signaling by providing a better balance between ones and zeroes. The extra code symbols are also used for control purposes, such as signaling the start-of-frame and end-of-frame, as well as carrier extension and error signaling.

Depending on the media system, the block encoded symbols may be transmitted using a simple two-level signaling system, or more complex multi-level line signaling. Multi-level line signaling can transmit more than one bit of information at a time. This effectively compresses the number of bits being transmitted into a smaller number of signal transitions on the cable. Using more complex line signaling schemes results in signal transition rates that can be supported on typical twisted-pair cable, which has limits on how fast it can transmit data.

The 100BASE-TX and 100BASE-FX Fast Ethernet systems (collectively known as 100BASE-X) both use the same block encoding scheme. The 100BASE-T4 and 100BASE-T2 systems use different encoding schemes and physical line signaling to provide Fast Ethernet signaling over twisted-pair cable that is lower quality and not rated to meet Category 5 specifications. However, since Category 5 cabling is widely adopted, there is no need for these two systems, and they never caught on in the marketplace; therefore, they will not be discussed here.

100BASE-X encoding

Rather than re-invent the wheel when it came to sending signals at 100 Mbps, the 100BASE-X media systems adopted portions of the block encoding and physical line signaling originally developed for the ANSI Fiber Distributed Data Interface (FDDI) network standard. FDDI is a 100 Mbps Token Ring network that was popular in the early 1990s. The block encoding used in FDDI and the 100BASE-X media types relies on a system called *4B/5B*, which divides the data into 4-bit blocks. The 4-bit blocks are translated into 5-bit code symbols for transmission over the media system. The encoded symbols are transmitted over fiber optic cables as two-level signals. The addition of a fifth bit means that the 100 Mbps data stream becomes a 125 Mbaud stream of signals on a fiber optic media system.*

The 5-bit encoding scheme allows for the transmission of 32 5-bit symbols, including 16 symbols that carry the four-bit data values from zero through F, along with a further set of 16 symbols used for control and other purposes. These other

* A *baud* is a unit of signaling speed per second. Mbaud = one million baud.

symbols include the IDLE symbol which is continually sent when no other data is present.* For this reason, the signaling system used in Fast Ethernet is continually active, sending IDLE symbols at 125 Mbaud if nothing else is going on.

Table 6-1 shows 9 of the 32 5-bit symbols that can be sent over the channel. Each five-bit symbol transmitted on the channel is mapped to a four-bit nibble of data, which is sent over the MII interface. There are a number of non-data symbols, some of which are given letter names (*J* and *K*) and used for special purposes such as indicating the start of preamble in a frame.

Table 6-1. Five-Bit Symbols

Five-Bit Code Group as Sent on the Channel	Data on MII interface	Interpretation
11110	0000	Data 0
01001	0001	Data 1
10100	0010	Data 2
10101	0011	Data 3
01010	0100	Data 4
Etc.	Etc.	Etc.
11101	1111	Data F
11111	Undefined	IDLE
11000	0101	*J*, always paired with *K*
10001	0101	*K*, always paired with *J*
Etc.	Etc.	Etc.

Carrier detection in MII transceivers ignores the IDLE symbols, so the carrier detect signal only becomes active when actual frame data symbols are seen on the channel. The pair of symbols called *J* and *K* are used together to indicate the start of the preamble in an Ethernet frame. A further pair of symbols called *T* and *R* are used to indicate the end of frame. The MII transceiver (PHY) deals with the task of recognizing these five-bit symbols, removing the special symbols, and delivering standard Ethernet frame data to the station interface or repeater port.

Each Fast Ethernet media system uses a different line signaling scheme to send the block encoded signals over the physical media. The details of each Fast Ethernet physical line signaling system are described in the individual media chapters.

* "IDLE" is not an acronym. Instead, IDLE is in uppercase in the Ethernet standard to show that the word is formally defined. The IDLE symbol is used to keep the signaling system active when there is no other data to send.

GMII Signal Encoding

The signaling techniques developed for the Fibre Channel standard and the twisted-pair Fast Ethernet media systems have been adapted and extended for Gigabit Ethernet. The fiber optic versions of Gigabit Ethernet are based on block encoding originally developed for the Fibre Channel standard. The details of this signal encoding are described in Chapter 12, *Gigabit Ethernet Fiber Optic Media System (1000BASE-X).*

Development of 1000BASE-T twisted-pair Gigabit Ethernet led to another set of block encoding and line signaling techniques based on work done for Fast Ethernet media systems. The following is an example of GMII block encoding for 1000BASE-T Ethernet.

Sending 1 billion bits per second over a twisted-pair media system is a daunting engineering challenge, and pushes the state of the art when it comes to signal encoding and transceiver chips. The 1000BASE-T system uses a block encoding scheme called 4D-PAM5, which transmits signals over four wire pairs. This coding scheme translates an 8-bit byte of data into a simultaneous transmission of four code symbols (4D) over four pairs of wire. The code symbols are sent over the media system using 5-level Pulse Amplitude Modulated (PAM5) signals. The five-level line signaling system includes an error correction signal to improve the signal to noise ratio on the cable.

The block encoding scheme and the line signaling systems used for Gigabit Ethernet media are quite complex, and of primary interest to designers of transceiver chips. Anyone who wants to examine the details can find the complete block encoding and line signaling specifications listed in many pages of dense engineering detail in the Gigabit Ethernet standard.

Ethernet Network Interface Card

In the earliest days of Ethernet, the NIC was a fairly large board covered with chips wired together to implement the necessary functions. Nowadays, an Ethernet interface is typically contained in a single chip that incorporates the required functions, including the MAC protocol. Ethernet interface chips are designed to keep up with the full rate of the Ethernet system.

However, the Ethernet interface is only one of an entire set of entities that must interact to make network services happen. Various elements have an effect on how many Ethernet frames a given computer can send and receive within a specified period of time. These include the speed with which your computer system can respond to signals from the Ethernet interface chip and the amount of available buffer memory for storing frames. The efficiency of the interface driver software also has an effect.

This is an important point to understand. For example, all Ethernet interface chips are capable of sending and receiving a frame at the full frame rate for the media system they support. However, the total performance of the computer system, buffer memory and software that interacts with the Ethernet interface is not specified in any standard. These days, most computers are more than capable of sending and receiving a constant stream of Ethernet frames at the maximum frame rate of a 10 or 100 Mbps Ethernet system.

Slower computers with lower performance Ethernet interfaces and insufficient buffer memory may not be able to keep up with the full frame rate. When Ethernet frames are not acknowledged and read by the computer, they are simply dropped by the interface. This is acceptable behavior as far as the standard is concerned, since no attempt is made to standardize computer performance.

Ethernet Interface Buyer's Guide

The Ethernet NICs are built into most desktop computers these days, so you probably won't have to concern yourself with buying one. A computer or workstation with built-in Ethernet is designed to get the best possible performance from that interface. The manufacturer of the computer will have done all the work of integrating the interface hardware and software. The network software loaded on the computer will already know how to work with the built-in interface.

However, some computers do not come with built-in Ethernet, requiring you to purchase a NIC separately. If you have to add an Ethernet card to a computer, the task can be fairly complex. It's up to you to find the right Ethernet card to meet your needs. There are buyer's guides and test reports in the major computer magazines (both in print and online), which evaluate the performance of various cards to help you make an informed decision. The decision can be difficult, since there are many Ethernet cards for sale at a wide range of prices and performance in terms of bus speed, buffer memory, and so on. When you buy an add-on interface, you need to determine which interface will work well with the network software you intend to run, as well as with your existing computer hardware.

These days you can buy interface cards that come with circuitry capable of operating at 10-, 100-, and 1000 Mbps, which often support multiple speeds in various combinations. These are referred to in vendor literature as 10/100 cards, 100/1000 cards, or even 10/100/1000 cards depending on the speeds supported. Multi-speed cards typically support the Auto-Negotiation standard, which allows the cards to automatically configure themselves for operation at the correct speed (see Chapter 5, *Auto-Negotiation*). Another option that is typically supported by multi-speed NICs is full-duplex operation. A useful feature to look for is the presence of status lights on the Ethernet card to indicate transmit and receive data, collisions,

the status of the link test signal, and so on. These lights are useful when troubleshooting a network connection.

Gigabit Ethernet interfaces

Many computers are currently using a significant percentage of their CPU power to generate heavy loads on a Fast Ethernet channel. These machines will not suddenly be able to go ten times faster just because they are connected to a Gigabit Ethernet channel. Gigabit Ethernet speeds and frame rates push the limits of what is possible today. As of this writing, most desktop computers and even many high performance server computers cannot keep up with the full frame rate of a Gigabit Ethernet channel.

Current PCI computer buses appear to have enough bandwidth to make significant use of a Gigabit Ethernet channel. However, interfaces will require special design to help speed up network protocol software, and to improve data rates in and out of a Gigabit Ethernet interface. Some vendors provide interfaces which provide high-level protocol packet processing onboard, to speed the flow of packets between the computer and the network. Another approach involves buffering several packets before interrupting the computer's CPU.