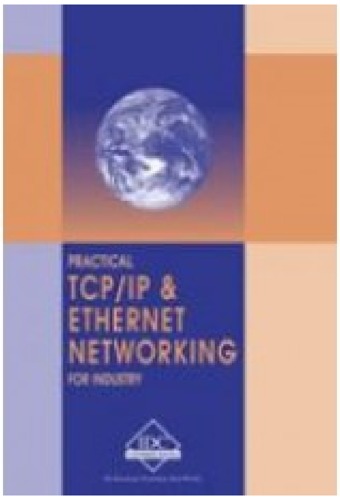
IDC Technologies - Books - 1031 Wellington Street West Perth WA 6005

Phone: +61 8 9321 1702 - Email: books@idc-online.com

TC-E - Practical TCPIP & Ethernet Networking for Industry



Price: $139.94

Ex Tax: $127.22

Short Description

SCADA systems, Programmable Logic Controllers and even low level instruments are using TCP/IP and Ethernet to transfer information. The reasons are not hard to find. TCP/IP and Ethernet are truly open standards available to competing manufacturers and providing the user with a common standard. In addition, the cost of TCP/IP and Ethernet is low. This manual has been structured to cover the main areas of TCP/IP and Ethernet in detail, while going through the practical implementation of TCP/IP in computer and industrial areas and practical use of the Internet and intranets. Troubleshooting and maintenance of TCP/IP networks and communication systems in an office and industrial environment are also covered.

Description

SCADA systems, Programmable Logic Controllers and even low level instruments are using TCP/IP and Ethernet to transfer information. The reasons are not hard to find. TCP/IP and Ethernet are truly open standards available to competing manufacturers and providing the user with a common standard. In addition, the cost of TCP/IP and Ethernet is low. This manual has been structured to cover the main areas of TCP/IP and Ethernet in detail, while going through the practical implementation of TCP/IP in computer and industrial areas and practical use of the Internet and intranets. Troubleshooting and maintenance of TCP/IP networks and communication systems in an office and industrial environment are also covered.

Table of Contents
Download Chapter List

Table of Contents

First Chapter
Introduction to Communications

1 Introduction to Communications

Learning objectives

When you have completed study of this chapter you should:

- Understand the main elements of the data communication process
- Understand the difference between analog and digital transmission
- Explain how data transfer is affected by attenuation, bandwidth and noise in the channel
- Comprehend the importance of synchronization of digital data systems
- Describe the basic synchronization concepts used with asynchronous and synchronous systems
- Explain the following types of encoding:
- Manchester
- RZ
- NRZ
- MLT-3
- 4B/5B
- Describe the basic error detection principles.

1.1 Data communications

Communications systems transfer messages from one location to another. The information component of a message is usually known as data (derived from the Latin word for items of information). All data is made up of unique code symbols or other entities on which the sender and receiver of the messages have agreed. For example, binary data is represented by two states viz. ‘0’ and ‘1’. These are referred to as *b*inary digi*ts* or ‘bits’ and are represented inside computers by the level of the electrical signals within storage elements; a high level could represent a ‘1’, and a low-level could represent a ‘0’. Alternatively, the data may be represented by the presence or absence of light in an optical fiber cable.

1.2 Transmitters, receivers and communication channels

A communications process requires the following components:

- A source of the information
- A transmitter to convert the information into data signals compatible with the communications channel
- A communications channel
- A receiver to convert the data signals back into a form the destination can understand
- The destination of the information

This process is shown in Figure 1.1.

Figure 1.1

Communications process

The transmitter encodes the information into a suitable form to be transmitted over the communications channel. The communications channel moves this signal from the source to one or more destination receivers. The channel may convert this energy from one form to another, such as electrical to optical signals, whilst maintaining the integrity of the information so the recipient can understand the message sent by the transmitter.

For the communications to be successful the source and destination must use a mutually agreed method of conveying the data.

The main factors to be considered are:

- The form of signaling and the magnitude(s) of the signals to be used
- The type of communications link (twisted pair, coaxial, optic fiber, radio etc)
- The arrangement of signals to form character codes from which the

message can be constructed
- The methods of controlling the flow of data
- The procedures for detecting and correcting errors in the transmission

The form of the physical connections is defined by interface standards. Some agreed-upon coding is applied to the message and the rules controlling the data flow and the detection and correction of errors are known as the **protocol**.

1.2.1 Interface standards

An interface standard defines the electrical and mechanical aspects of the interface to allow the communications equipment from different manufacturers to interoperate.

A typical example is the TIA-232-F interface standard (commonly known as RS-232). This specifies the following three components:

- **Electrical signal characteristics**– defining the allowable voltage levels, grounding characteristics etc
- **Mechanical characteristics**– defining the connector arrangements and pin assignments
- **Functional description of the interchange circuits**– defining the function of the various data, timing and control signals used at the interface

It should be emphasized that the interface standard only defines the electrical and mechanical aspects of the interface between devices and does not define how data is transferred between them.

1.2.2 Coding

A wide variety of codes have been used for communications purposes. Early telegraph communications used Morse code with human operators as transmitter and receiver. The Baudot code introduced a constant 5-bit code length for use with mechanical telegraph transmitters and receivers. The commonly used codes for data communications today are the Extended Binary Coded Decimal Interchange Code (EBCIDIC) and the American Standard Code for Information Interchange (ASCII).

1.2.3 Protocols

A protocol is essential for defining the common message format and procedures for transferring data between all devices on the network. It includes the following important features:

- **Initialization:**Initializes the protocol parameters and commences the data transmission
- **Framing and synchronization:**Defines the start and end of the frame and how the receiver can synchronize to the data stream
- **Flow control:**Ensures that the receiver is able to advise the transmitter to regulate the data flow and ensure no data is lost.
- **Line control:**Used with half-duplex links to reverse the roles of transmitter and receiver and begin transmission in the other direction.
- **Error control:**Provides techniques to check the accuracy of the received data to identify transmission errors. These include block redundancy checks and cyclic redundancy checks
- **Time out control:** Procedures for transmitters to retry or abort transmission when acknowledgments are not received within agreed time limits

1.2.4 Some commonly used communications protocols

- X/ Y/ Z modem and Kermit for asynchronous file transmission
- Binary Synchronous Protocol (BSC), Synchronous Data Link Control (SDLC) or High-Level Data Link Control (HDLC) for synchronous transmissions
- Industrial protocols such as Modbus and DNP3

1.3 Types of communication channels

An analog communications channel conveys signals that change continuously in both frequency and amplitude. A typical example is a sine wave as illustrated in Figure 1.2. On the other hand, digital transmission employs a signal of which the amplitude varies between a few discrete states. An example is shown in Figure 1.3.

Figure 1.2

Analog signal

Figure 1.3

Digital signal

1.4 Communications channel properties

1.4.1 Signal attenuation

As the signal travels along a communications channel its amplitude decreases as the physical medium resists the flow of the signal energy. This effect is known as signal attenuation. With electrical signaling some materials such as copper are very efficient conductors of electrical energy. However, all conductors contain impurities that resist the movement of the electrons that constitute the electric current. The resistance of the conductors causes some of the electrical energy of the signal to be converted to heat as the signal progresses along the cable resulting in a continuous decrease in the electrical signal. The signal attenuation is measured in terms of signal loss per unit length of the cable, typically dB/km.

To allow for attenuation, a limit is set for the maximum length of the communications channel. This is to ensure that the attenuated signal arriving at the receiver is of sufficient amplitude to be reliably detected and correctly interpreted. If the channel is longer than this maximum specified length, repeaters must be used at intervals along the channel to restore the signal to acceptable levels.

Figure 1.4

Signal repeaters

Signal attenuation increases as the frequency increases. This causes distortion to practical signals containing a range of frequencies. This problem can be overcome by the use of amplifiers that amplify the higher frequencies by greater amounts.

1.4.2 Channel bandwidth

The quantity of information a channel can convey over a given period is determined by its ability to handle the rate of change of the signal, i.e. the signal frequency. The bandwidth of an analog channel is the difference between the highest and lowest frequencies that can be reliably transmitted over the channel. These frequencies are often defined as those at which the signal at the receiving end has fallen to half the power relative to the mid-band frequencies (referred to as the -3 dB points), in which case the bandwidth is known as the -3 dB bandwidth.

Figure 1.5

Channel bandwidth

Digital signals are made up of a large number of frequency components, but only those within the bandwidth of the channel will be able to be received. It follows that the larger the bandwidth of the channel, the higher the data transfer rate can be and more high frequency components of the digital signal can be transported, and so a more accurate reproduction of the transmitted signal can be received.

Figure 1.6

Effect of channel bandwidth on digital signal

The maximum data transfer rate (C) of the transmission channel can be determined from its bandwidth, by use of the following formula derived by Shannon.

$C = 2B \log_2 M$ bps

Where:

B = bandwidth in hertz and M levels are used for each signaling element.

In the special case where only two levels, ‘ON’ and ‘OFF’ are used (binary), $M = 2$ and $C = 2B$. For example, the maximum data transfer rate for a PSTN channel with 3200 hertz bandwidth carrying a binary signal would be 2 ´ 3200 = 6400 bps. The achievable data transfer rate would then be reduced by half because of the Nyquist rate. It is further reduced in practical situations because of the presence of noise on the channel to approximately 2400 bps unless some modulation system is used.

1.4.3 Noise

As the signals pass through a communications channel the atomic particles and molecules in the transmission medium vibrate and emit random electromagnetic signals as noise. The strength of the transmitted signal is normally large relative to the noise signal. However, as the signal travels through the channel and is attenuated, its level can approach that of the noise. When the wanted signal is not significantly higher than the background noise, the receiver cannot separate the data from the noise and communication errors occur.

An important parameter of the channel is the ratio of the power of the received signal (S) to the power of the noise signal (N). The ratio S/N is called the Signal to Noise ratio, normally expressed in decibels (dB).

$S/N = 10 \log_{10} (S/N)$ dB

Where signal and noise levels are expressed in watts, or

$S/N = 20 \log_{10} (S/N)$ dB

Where signal and noise levels are expressed in volts

A high signal to noise ratio means that the wanted signal power is high compared to the noise level, resulting in good quality signal reception

The theoretical maximum data transfer rate for a practical channel can be calculated using the Shannon-Hartley law, which states that:

$C = B \log_2 (1+S/N)$ bps

Where:

C = data rate in bps

B = bandwidth of the channel in hertz

S = signal power in watts and

N = noise power in watts

It can be seen from this formula that increasing the bandwidth or increasing the S/N ratio will allow increases to the data rate, and that a relatively small increase in bandwidth is equivalent to a much greater increase in S/N ratio.

Digital transmission channels make use of higher bandwidths and digital repeaters or regenerators to regenerate the signals at regular intervals and maintain acceptable signal to noise ratios. The degraded signals received at the regenerator are detected, then re-timed and retransmitted as nearly perfect replicas of the original digital signals, as shown in Figure 1.7. Provided the signal to noise ratios is maintained in each link, there is no accumulated noise on the signal, even when transmitted over thousands of kilometers.

Figure 1.7

Digital link

1.5 Data transmission modes

1.5.1 Direction of signal flow

Simplex

A simplex channel is unidirectional and allows data to flow in one direction only, as shown in Figure 1.8. Public radio broadcasting is an example of a simplex transmission. The radio station transmits the broadcast program, but does not receive any signals back from the receiver(s).

Figure 1.8

Simplex transmission

This has limited use for data transfer purposes, as we invariably require the flow of data in both directions to control the transfer process, acknowledge data etc.

Half-duplex

Half-duplex transmission allows simplex communication in both directions over a single channel, as shown in Figure 1.9. Here the transmitter at ‘A’ sends data to a receiver at ‘B’. A line turnaround procedure takes place whenever transmission is required in the opposite direction. Transmitter ‘B’ is then enabled and communicates with receiver ‘A’. The delay in the line turnaround reduces the available data throughput of the channel.

Figure 1.9

Half-duplex transmission

Full-duplex

A full-duplex channel gives simultaneous communications in both directions, as shown in Figure 1.10.

Figure 1.10

Full-duplex transmission

1.5.2 Synchronization of digital data signals

Data communications depends on the timing of the signal transmission and reception being kept correct throughout the message transmission. The receiver needs to look at the incoming data at correct instants before determining whether a ‘1’ or ‘0’ was transmitted. The process of selecting and maintaining these sampling times is called synchronization.

In order to synchronize their transmissions, the transmitting and receiving devices need to agree on the length of the code elements to be used, known as the bit time. The receiver also needs to synchronize its clock with that of the sender in order to determine the right times at which to sample the data bits in the message.   A device at the receiving end of a digital channel can synchronize itself using either asynchronous or synchronous means as outlined below.

1.5.3 Asynchronous transmission

Here the transmitter and receiver operate independently, and the receiver synchronizes its clock with that of the transmitter (only) at the start of each message frame. Transmissions are typically around one byte in length, but can also be longer. Often (but not necessarily) there is no fixed relationship between one message frame and the next, such as a computer keyboard input with potentially long random pauses between keystrokes.

Figure 1.11
Asynchronous data transmission

At the receiver the channel is monitored for a change in voltage level. The leading edge of the start bit is used to set a bistable multivibrator (flip-flop), and half a bit period later the output of the flip-flop is compared (ANDed) with the incoming signal. If the start bit is still present, i.e. the flip-flop was set by a genuine start bit and not by a random noise pulse, the bit clocking mechanism is set in motion. In fixed-rate systems this half-bit delay can be generated with a monostable (one-shot) multivibrator, but in variable-rate systems it is easier to feed a 4-bit (binary or BCD) counter with a frequency equal to 16 times the data frequency and observing the ‘D’ (most significant bit) output, which changes from 0 to 1 at a count of eight, half-way through the bit period.

Figure 1.12

Clock synchronization at the receiver

The input signal is fed into a serial-in parallel-out shift register. The data bits are then captured by clocking the shift register at the data rate, in the middle of each bit. For an eight-bit serial transmission (data plus parity), this sampling is repeated for each of the eight data bits and a final sample is made during the ninth time interval to identify the stop bit and to confirm that the synchronization has been maintained to the end of the frame. Figure 1.13 illustrates the asynchronous data reception process.

Figure 1.13

Asynchronous data reception (clock too slow)

1.5.4 Synchronous transmission

The receiver here is initially synchronized with the transmitter, and then maintains this synchronization throughout the continuous transmission of multiple bytes. This is achieved by special data coding schemes, such as Manchester, which ensure that the transmitted clock is encoded into the transmitted data stream. This enables the synchronization to be maintained at the receiver right to the last bit of the message, and thus allows larger frames of data (up to several thousand bytes) to be efficiently transferred at high data rates.

The synchronous system packs many bytes together and sends them as a continuous stream, called a frame. Each frame consists of a header, data field and checksum. Clock synchronization is provided by a predefined sequence of bits called a preamble.

In some cases the preamble is terminated with a ‘Start of Frame Delimiter’ or SFD. Some systems may append a post-amble if the receiver is not otherwise able to detect the end of the message. An example of a synchronous frame is shown in Figure 1.14. Understandably all high-speed data transfer systems utilize synchronous transmission systems to achieve fast, accurate transfers of large amounts of data.

Figure 1.14

Synchronous frame

1.6 Encoding methods

1.6.1 Manchester

Manchester is a bi-phase signal-encoding scheme, and is used in the older 10 Mbps Ethernet LANs. The direction of the transition in mid-interval (negative to positive or positive to negative) indicates the value (‘1’ or ‘0’, respectively) and provides the clocking.

The Manchester codes have the advantage that they are self-clocking. Even a sequence of one thousand ‘0s’ will have a transition in every bit; hence the receiver will not lose synchronization. The price paid for this is a bandwidth requirement double that which is required by the RZ-type methods.

The Manchester scheme follows these rules:

- +V and –V voltage levels are used
- There is a transition from one to the other voltage level halfway through each bit interval
- There may or may not be a transition at the start of each bit interval, depending on whether the bit value is a ‘0’ or ‘1’
- For a ‘1’ bit, the transition is always from a –V to +V; for a ‘0’ bit, the transition is always from a +V to a –V

In Manchester encoding, the beginning of a bit interval is used merely to set the stage. The activity in the middle of each bit interval determines the bit value: upward transition for a ‘1’ bit, downward for a ‘0’ bit.

1.6.2 Differential Manchester

Differential Manchester is a bi-phase signal-encoding scheme used in Token Ring LANs. The presence or absence of a transition at the beginning of a bit interval indicates the value; the transition in mid-interval just provides the clocking.

For electrical signals, bit values will generally be represented by one of three possible voltage levels: positive (+V), zero (0 V), or negative (–V). Any two of these levels are needed – for example, + V and –V.

There is a transition in the middle of each bit interval. This makes the encoding method self-clocking, and helps avoid signal distortion due to DC signal components.

For one of the possible bit values but not the other, there will be a transition at the start of any given bit interval. For example, in a particular implementation, there may be a signal transition for a ‘1’ bit.

In differential Manchester encoding, the presence or absence of a transition at the beginning of the bit interval determines the bit value. In effect, ‘1’ bits produce vertical signal patterns; ‘0’ bits produce horizontal patterns. The transition in the middle of the interval is just for timing.

1.6.3 RZ (return to zero)

The RZ-type codes consume only half the bandwidth taken up by the Manchester codes. However, they are not self-clocking since a sequence of a thousand ‘0s’ will result in no movement on the transmission medium at all.

RZ is a bipolar signal-encoding scheme that uses transition coding to return the signal to a zero voltage during part of each bit interval. It is self-clocking.

In the differential version, the defining voltage (the voltage associated with the first half of the bit interval) changes for each ‘1’ bit, and remains unchanged for each ‘0’ bit.

In the non-differential version, the defining voltage changes only when the bit value changes, so that the same defining voltages are always associated with ‘0’ and ‘1’. For example, +5 volts may define a ‘1’, and –5 volts may define a ‘0’.

1.6.4 NRZ (non-return to zero)

NRZ is a bipolar encoding scheme. In the non-differential version it associates, for example, +5 V with ‘1’ and –5 V with ‘0’.

In the differential version, it changes voltages between bit intervals for ‘1’ values but not for ‘0’ values. This means that the encoding changes during a transmission. For example, ‘0’ may be a positive voltage during one part and a negative voltage during another part, depending on the last occurrence of a ‘1’. The presence or absence of a transition indicates a bit value, not the voltage level.

1.6.5 MLT-3

MLT-3 is a three-level encoding scheme that can also scramble data. This scheme is one proposed for use in FDDI networks. The MLT-3 signal-encoding scheme uses three voltage levels (including a zero level) and changes levels only when a ‘1’ occurs.
It follows these rules:

- +V, 0 V, and –V voltage levels are used
- The voltage remains the same during an entire bit interval; that is, there are no transitions in the middle of a bit interval
- The voltage level changes in succession; from +V to 0 V to –V to 0 V to +V, and so on
- The voltage level changes only for a ‘1’ bit

MLT-3 is not self-clocking, so that a synchronization sequence is needed to make sure the sender and receiver are using the same timing.

1.6.6 4B/5B

The Manchester codes, as used for 10 Mbps Ethernet, are self-clocking but consume unnecessary bandwidth (at 10 Mbps it introduces a 20 MHz frequency component on the medium). For this reason it is not possible to use it for 100 Mbps Ethernet, even over Cat5 cable. A solution is to revert back to one of the more bandwidth efficient methods such as NRZ or RZ. The problem with these, however, is that they are not self-clocking and hence the receiver loses synchronization if several zeros are transmitted sequentially. This problem, in turn, is overcome by using the 4B/5B technique.

The 4B/5B technique codes each group of four bits into a five-bit code. For example, the binary pattern 0110 is coded into the five-bit pattern 01110. This code table has been designed in such a way that no combination of data can ever be encoded with more than 3 zeros on a row. This allows the carriage of 100 Mbps data by transmitting at 125 MHz, as opposed to the 200 Mbps required by Manchester encoding.

Table 1.1

4B/5B data coding

1.7 Error detection

All practical data communications channels are subject to noise, particularly where equipment is situated in industrial environments with high electrical noise, such as electromagnetic radiation from adjacent equipment or electromagnetic induction from adjacent cables. As a consequence the received data may contain errors. To ensure reliable data communication we need to check the accuracy of each message.

Asynchronous systems often use a single bit checksum, the parity bit, for each message, calculated from the seven or eight data bits in the message. Longer messages require more complex checksum calculations to be effective. For example the Longitudinal Redundancy Check (LRC) calculates an additional byte covering the content of the message (up to 15 bytes) while a two-byte arithmetic checksum can be used for messages up to 50 bytes in length. Most high-speed LANs use a 32-bit Cyclic Redundancy Check (CRC)

The CRC method detects errors with very high degree of accuracy in messages of any length and can, for example, detect the presence of a single bit error in a frame containing tens of thousands of bits of data. The CRC treats all the bits of the message block as one binary number that is then divided by a known polynomial. For a 32-bit CRC this is a specific 32-bit number, specially chosen to detect very high percentages of errors, including all error sequences of less than 32 bits. The remainder found after this division process is the CRC. Calculation of the CRC is carried out by the hardware in the transmission interface of LAN adapter cards.