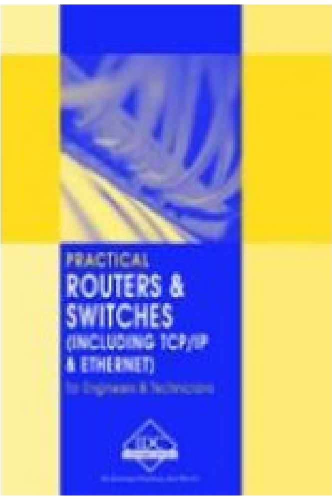
IDC Technologies - Books - 1031 Wellington Street West Perth WA 6005

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

RS-E - Routers & Switches (TCPIP & Ethernet)



Price: $139.94

Ex Tax: $127.22

Short Description

This manual goes through the basics of routers, routed and routing protocols and the basic rules to follow in building internet works. If you are using any form of communication system or are applying modern PLC's/SCADA systems this manual will give you the essential tools in working with your networks.

Description

This manual goes through the basics of routers, routed and routing protocols and the basic rules to follow in building internet works. If you are using any form of communication system or are applying modern PLC's/SCADA systems this manual will give you the essential tools in working with your networks.

Table of Contents

Download Chapter List

Table of Contents

First Chapter
Introduction to Communications

1

Introduction to communications

Objectives

When you have completed study of this chapter you should be able to:

- 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
- Know 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 exist to transfer information from one location to another. The components of the information or message are usually known as data (derived from the Latin word for items of information). All data are 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 "0" and "1". These are referred to as Binary digits or "bits".

These bits are represented inside our computers by the level of the electrical signals within storage elements; a high level could represent a "1", and a low-level 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

Communication process

The transmitter encodes the information into a suitable form to be transmitted over the communications channel. The communications channel moves this signal as electromagnetic energy 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 coding is applied to the message and the rules controlling the data flow and 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 operate together.

A typical example is the **EIA/TIA-232-E** interface standard. 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 cover how data is transferred between them (see Table 1.1).

Table 1.1
ASCII code table

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 Standards Committee 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

- Xmodem or 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 Manufacturing Automation Protocol (MAP), Technical Office Protocol (TOP), Modbus, Data Highway Plus, HART, Profibus, Foundation Fieldbus, etc

1.3 Types of communication channels

1.3.1 Analog communications channels

An analog communications channel conveys analog signals that are changing continuously in both frequency and amplitude. These signals are commonly used for audio and video communication as illustrated in Figure 1.2 and Figure 1.3.

Figure 1.2

Analog signal

1.3.2 Digital communications channels

Figure 1.3

Digital signal

1.4 Communications channel properties

The physical properties of the communications channels limit their ability to carry information in either analogue or digital form. The principal effects are signal attenuation, channel bandwidth and noise.

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 electromagnetic 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 energy 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 (see Figure 1.4).

Figure 1.4

Signal attenuation

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 length, amplifiers or repeaters must be used at intervals along the channel to restore the signal to acceptable levels (see Figure 1.5).

Figure 1.5

Signal repeaters

Signal attenuation increases as the frequency increases. This causes distortion to practical signals containing a range of frequencies. This is illustrated in Figure 1.4 where the rise-times of the attenuated signals progressively decrease as the signal travels through the channel, caused by the greater attenuation of the high frequency components. 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, that is its frequency. An analog signal varies between a minimum and maximum frequency and the difference between those frequencies is the bandwidth of that signal. The bandwidth of an analog channel is the difference between the highest and lowest frequencies that can be reliably received over the channel. These frequencies are often those at which the signal has fallen to half the power relative to the mid band frequencies, referred to as 3dB points. In this case the bandwidth is known as the 3dB bandwidth (see Figure 1.6).

Figure 1.6

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 (see Figure 1.7).

Figure 1.7

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 = 2 B. As an example, the maximum data transfer rate for a PSTN channel of bandwidth 3200 Hertz carrying a binary signal would be 2 x 3200 = 6400 bps. The achievable data transfer rate is reduced to ½ of 6400 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, which is normally expressed in decibels, abbreviated to dB.

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

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:

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

Where

C data rate in bps

B bandwidth of the channel in Hertz

S signal power in watts and N is the noise power in watts

It can be seen from this formula that increasing the bandwidth or increasing the signal to noise ratio will allow increases to the data rate, and that a relatively small increase in bandwidth is equivalent to a much greater increase in signal to noise 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 1.8. Provided the signal to noise ratios are maintained in each link, there is no accumulated noise on the signal, even when transmitted thousands of kilometers.

Figure 1.8

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 1.9. 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 your radio receiver.

Figure 1.9

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 us to provide simplex communication in both directions over a single channel, as shown in Figure 1.10. Here the transmitter at station "A" sends data to a receiver at station "B". A line turnaround procedure takes place whenever transmission is required in the opposite direction. The station "B" transmitter is then enabled and communicates with the receiver at station "A". The delay in the line turnaround procedures reduces the available data throughput of the communications channel.

Figure 1.10

Half-duplex transmission

Full-duplex

A Full-duplex channel gives simultaneous communications in both directions, as shown in Figure 1.11.

Figure 1.11

Full duplex transmission

1.5.2 Synchronization of digital data signals

Data communications depends on the timing of the signal generation and reception being kept correct throughout the message transmission. The receiver needs to look at the incoming data at the 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 needs to extract the transmitted clock signal encoded into the received data stream. By synchronizing the bit time of the receiver's clock with that encoded by the sender, the receiver is able to determine the right times to detect the data transitions in the message and correctly receive the message. The devices at both ends of a digital channel can synchronize themselves using either asynchronous or synchronous transmission as outlined below.

1.5.3 Asynchronous transmission

Here the transmitter and receiver operate independently, and exchange a synchronizing pattern at the start of each message code element (frame). 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 (see Figure 1.12).

Figure 1.12
Asynchronous data transmission

At the receiver the channel is sampled at a high rate, typically in excess of 16 times the bit rate of the data channel, to accurately determine the centers of the synchronizing pattern (start bit) and its duration (bit time) (see Figure 1.13).

Figure 1.13
Clock estimation at receiver

The data bits are then determined by the receiver sampling the channel at intervals corresponding to the centers of each transmitted bit. These are estimated by delaying multiples of the bit time from the centers of the start bit. For an eight-bit serial transmission, this sampling is repeated for each of the eight data bits then a final sample is made during the ninth time interval. This sample is to identify the stop bit and confirm that the synchronization has been maintained to the end of the message frame. Figure 1.14 illustrates the asynchronous data reception process.

Figure 1.14

Asynchronous data reception

1.5.4 Synchronous transmission

The receiver here is initially synchronized to the transmitter then maintains this synchronization throughout the continuous transmission. This is achieved by special data coding schemes, such as Manchester Encoding, which ensure that the transmitted clock is continuously encoded into the transmitted data stream. This enables the synchronization to be maintained at any receiver right to the last bit of the message, which could be as large as 4500 bytes (36000 bits). This allows larger frames of data to be efficiently transferred at higher data rates. The synchronous system packs many characters together and sends them as a continuous stream, called a block. For each transmission block there is a preamble, containing the start delimiter for initial synchronization purposes and information about the block, and a post-amble, to give error checking, etc. An example of a synchronous transmission block is shown in Figure 1.15. Understandably all high-speed data transfer systems utilize synchronous transmission systems to achieve fast, accurate transfers of large blocks of data.

Figure 1.15

Synchronous transmission block

1.6 Encoding methods

1.6.1 Manchester

Manchester is a bi-phase signal-encoding scheme used in 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 "0's" 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 that 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 (0V), 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 "0's" 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, +5V with 1 and -5V 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, 0V, 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 0V to -V to 0V 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. For this reason, it is not possible to use it for 100 Mbps Ethernet over CAT5 cable. A solution to the problem 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 (see Table 1.2).

Table 1.2

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 an Arithmetic Checksum (calculates two additional bytes) can be used for messages up to 50 bytes in length. Most high speed Local Area Networks uses a 32-bit **Cyclic Redundancy Check** (CRC).

1.7.1 Data echoing

One of the simplest ways of checking data accuracy is for the receiver to send back the data to the transmitter for verification. This is only suitable for short messages and is not particularly efficient. Since the message has to be sent correctly in both directions there is no way of knowing whether any error occurred on either the forward or return message or both!

1.7.2 Checksum calculation

All of the sophisticated protocols utilize some form of error detection process whereby the sender of the message calculates some form of checksum from the data to be sent. This is appended to the end of the data and sent to the receiver. At the receiver the same calculation is repeated, using the data bits received, and the result is then compared with the checksum originally sent. If both checksums match the data is considered correct.

If a difference is detected then the data is assumed corrupt and a retransmission has to be requested. This process is illustrated in Figure 1.16.

Figure 1.16

Error detection concept

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 an Arithmetic Checksum (calculates two additional bytes) can be used for messages up to 50 bytes in length. Most high speed Local Area Networks uses a 32-bit Cyclic

Redundancy Check (CRC).

1.7.3 Cyclic Redundancy Check (CRC)

The **Cyclic Redundancy Check** (CRC) enables detection of errors with very high accuracy in messages of any length. So, for example, we can detect the presence of a single bit in error in a synchronous data frame containing 36,000 bits. The CRC works by treating 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 generator, 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.