appendix a CRC implementation

Although the requirement to perform multiple (modulo-2) divisions to compute a CRC may appear to be relatively complicated, in practice it can be done quite readily in either hardware or software. To illustrate this, a hardware implementation of the scheme used in Figure 6.22 is given in Figure A.1(a).

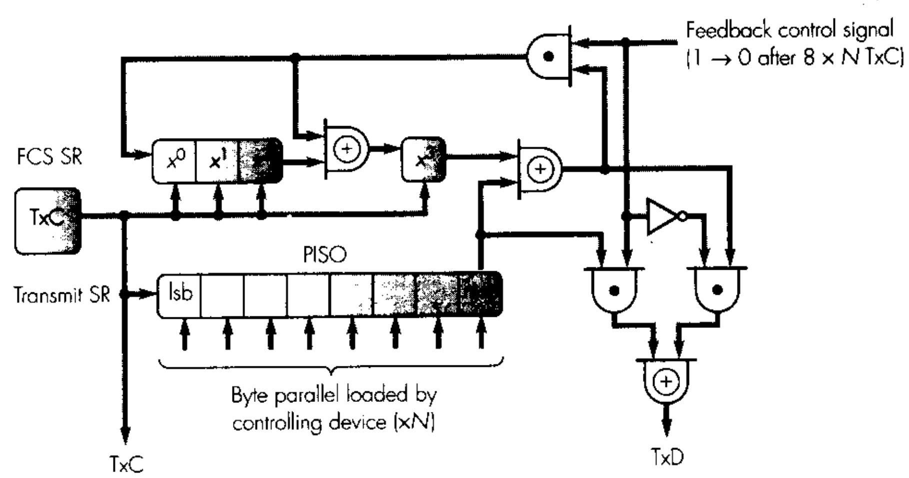
In this example, since we are to generate four FCS digits, we need only a 4-bit shift register to represent bits x^3, x^2, x^1, and x^0 in the generator polynomial. We often refer to these as the active bits of the generator. With this generator polynomial, digits x^3 and x^0 are binary 1 while digits x^2 and x^1 are binary 0. The new states of shift register elements x^1 and x^2 simply take on the states of x^0 and x^1 directly; the new states of elements x^0 and x^3 are determined by the state of the feedback path exclusive ORed with the preceding digit.

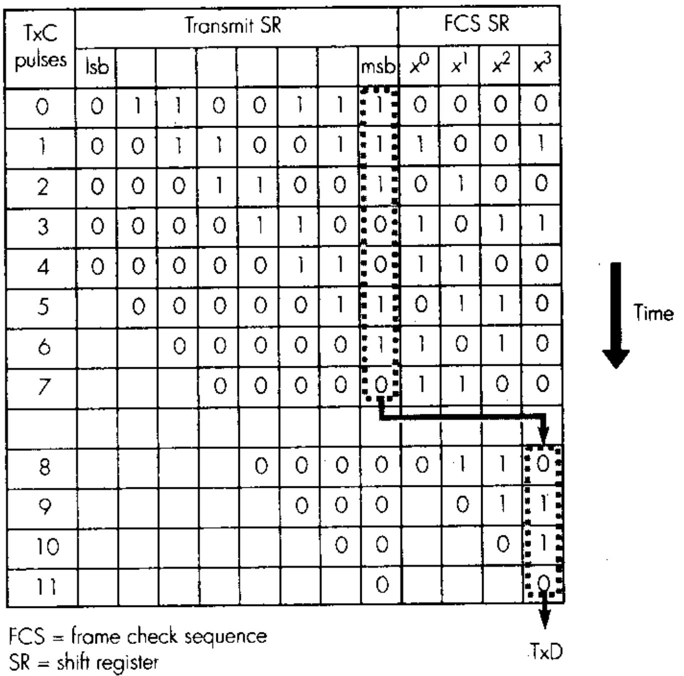
The circuit operates as follows. The FCS shift register is cleared and the first 8-bit byte in the frame is parallel-loaded into the PISO transmit shift register. This is then shifted out to the transmission line, most significant bit first, at a rate determined by the transmitter clock TxC. In time synchronism with this, the same bitstream is exclusive-ORed with x^3 and passed via the feedback path to the selected inputs of the FCS shift register. As each subsequent 8-bit byte is loaded into the transmit shift register and bit-serially transmitted to line, the procedure repeats. Finally, after the last byte in the frame has been output, the transmit shift register is loaded with zeros and the feedback control signal changes from 1 to 0 so that the current contents of the FCS shift register – the computed remainder – follow the frame contents onto the transmission line.

In Figure A.1(a) the contents of the transmit and FCS shift registers assume just a single-byte frame ($N = 1$), and hence correspond to the earlier example in Figure 6.22. In the figure the contents of both the transmit and FCS shift registers are shown after each shift (transmit clock) pulse. The transmitted bitstream is as shown in the hashed boxes.

The corresponding receiver hardware is similar to that used at the transmitter, as shown in Figure A.1(b). The received data (RxD) is sampled (shifted) into the SIPO receive shift register in the centre (or later with Manchester encoding) of the bit cell. Also, as before, in time synchronization with this the bitstream is exclusive-ORed with x^3 and fed into the FCS shift register. As each 8-bit byte is received, it is read by the controlling device. Again, the contents shown are for a frame comprising just a single byte of data.

(a)



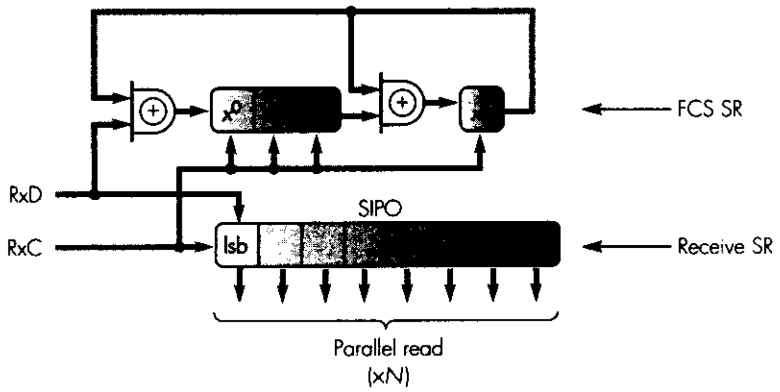


TxC pulses	Transmit SR								FCS SR			
	lsb							msb	x^0	x^1	x^2	x^3
0	0	1	1	0	0	1	1	1	0	0	0	0
1	0	0	1	1	0	0	1	1	1	0	0	1
2	0	0	0	1	1	0	0	1	0	1	0	0
3	0	0	0	0	1	1	0	0	1	0	1	1
4	0	0	0	0	0	1	1	0	1	1	0	0
5		0	0	0	0	0	1	1	0	1	1	0
6			0	0	0	0	0	1	1	0	1	0
7				0	0	0	0	0	1	1	0	0
8					0	0	0	0	0	1	1	0
9						0	0	0		0	1	1
10							0	0			0	1
11								0				0

FCS = frame check sequence
SR = shift register
PISO = parallel-in, serial-out

Figure A.1 CRC hardware implementation schematic: (a) CRC generation; (b) CRC checking.

(b)



RxC	RxD	Receive SR (shift register)								FCS SR			
		lsb							msb	x^0	x^1	x^2	x^3
0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1								1	0	0	0
2	1	1	1							1	1	0	0
3	0	1	1	1						1	1	1	0
4	0	0	1	1	1					0	1	1	1
5	1	0	0	1	1	1				1	0	1	0
6	1	1	0	0	1	1	1			1	1	0	1
7	0	1	1	0	0	1	1	1		0	1	1	1
8	0	0	1	1	0	0	1	1	1	1	0	1	0
		Byte read by controlling device											
9	1									0	1	0	1
10	1									0	0	1	1
11	0									0	0	0	0
12										0	0	0	0

Time ↓

SIPO = serial-in, parallel-out

Remainder = 0

Figure A.1 Continued.

The hardware in Figure A.1 is normally incorporated into the transmission control circuits associated with bit-oriented transmission. In some instances, however, a CRC is used in preference to a block sum check with character-oriented transmission. In such cases, the CRC must normally be generated in software by the controlling device rather than in hardware. This is relatively straightforward as we can see from the pseudocode in Figure A.2.

```
{Assume a preformatted frame to be transmitted (including a zero byte at its tail) or a received frame
is stored in a byte array buff[1.. count]. Also that the 8 active bits of a 9-bit divisor are stored in the
most-significant 8 bits of a 16-bit integer CRCDIV. The following function will compute and return the
8-bit CRC}

function  CRC : byte;
var       i, j : integer;
          data : integer

begin     data := buff[1] shl 8;
          for j := 2 to count do
               begin
                    data := data + buff [j];
                    for i := 1 to 8 do
                         if ((data and $8000) = $8000) then
                         begin data := data shl1;
                                data := data xor CRCDIV; end
                         else   data := data shl 1;
               end;

          CRC := data shr 8;
end;
```

Figure A.2 Pseudocode for the computation and checking of an 8-bit CRC.

The code assumes an 8-bit generator polynomial (divisor) and that the preformatted frame – STX, ETX, and so on – is stored in an array. The same code can be used for CRC generation and checking; for generation the array will contain a byte/character comprising all zeros at its tail.

appendix b Forward error control

B.1 Introduction

With an automatic repeat request (ARQ) error control scheme, additional check digits are appended to each transmitted message (frame) to enable the receiver to detect when an error is present in a received message, assuming certain types of error. If an error is detected, additional control procedures are used to request another copy of the message. With forward error control (FEC), sufficient additional check digits are added to each transmitted message to enable the receiver not only to detect the presence of one or more errors in a received message but also to locate the position of the error(s). Furthermore, since the message is in a binary form, correction is achieved simply by inverting the bit(s) that have been identified as erroneous.

In practice, the number of additional check digits required for error correction is much larger than that needed for just error detection. In most applications involving terrestrial (land-based) links, ARQ methods similar to those described in Chapter 6 are more efficient than FEC methods, and hence are the most frequently used. Such methods rely on a return path for acknowledgment purposes. However, in most entertainments applications, a return path is simply not available or, even if one was available, the round-trip delay associated with it may be very long compared with the data transmission rate of the link. For example, with many satellite links the propagation delay may be such that several hundred megabits may be transmitted by the sending station before an acknowledgment could be received in the reverse direction. In such applications, FEC methods are used. The aim of this appendix is to give an introduction to the techniques most widely used with FEC methods.

B.2 Block codes

An example of a block code is the Hamming single-bit code. In practice, this FEC method is of limited use for digital transmission. Nevertheless, we shall look at it briefly to introduce the subject of block codes and some of the terms associated with coding theory. Clearly, a comprehensive description of the subject of coding theory is beyond the scope of this book and hence the aim here is simply to give a brief introduction. If you have an interest in coding theory and would like to gain a more extensive coverage, consult some of the references given in the bibliography at the end of the book.

Recall that the term used in coding theory to describe the combined message unit, comprising the useful data bits and the additional check bits, is **codeword**. The minimum number of bit positions in which two valid codewords differ is known as the **Hamming distance** of the code. For example, consider a coding scheme that has seven data bits and a single parity bit per codeword. Assuming even parity is being used, consecutive codewords in this scheme are as follows:

```
0000000  0
0000001  1
0000010  1
0000011  0
```

We can see from this list that such a scheme has a Hamming distance of 2, as each valid codeword differs in at least two bit positions. This means that it does not detect 2-bit errors since the resulting (corrupted) bit pattern will be a different but valid codeword. However, it does detect all single-bit errors since, if a single bit in a codeword is corrupted, an invalid codeword will result.

In general, the error-detecting and error-correcting properties of a coding scheme are both related to its Hamming distance. It can be shown that to detect n errors, we must use a coding scheme with a Hamming distance of $n + 1$, while to correct for n errors, we must use a code with a Hamming distance of $2n + 1$.

The simplest error-correcting coding scheme is the Hamming single-bit code. Such a code detects not only when a single-bit error is present in a received codeword but also the position of the error. The corrected codeword is derived by inverting the identified erroneous bit. This type of code is known as a block code, since the original message to be transmitted is treated as a single block (frame) during the encoding and subsequent decoding processes. In general, with a block code, each block of k source digits is encoded to produce an n-digit block (n greater than k) of output digits. The encoder is said to produce an (n, k) code. The ratio k/n is known as the **code rate** or **code efficiency** while the difference $1 - k/n$ is known as the **redundancy**.

To illustrate this, consider a Hamming code to detect and correct for single-bit errors assuming each codeword contains a 7-bit data field – an ASCII character, for example. Such a coding scheme requires four check bits since, with this scheme, the check bits occupy all bit positions that are powers of 2. This code is known as an (11, 7) block code with a rate of 7/11 and a redundancy of 1 – 7/11. For example, the bit positions of the value 1001101 are as follows:

```
11 10 9 8 7 6 5 4 3 2 1
 1  0 0 x 1 1 0 x 1 x x
```

The four bit positions marked 'x' are used for the check bits, which are derived as follows. The 4-bit binary numbers corresponding to those bit positions with a binary 1 are added together using modulo-2 arithmetic and the four check bits are the following 4-bit sum:

$$\begin{array}{rl} 11 = & 1\,0\,1\,1 \\ 7 = & 0\,1\,1\,1 \\ 6 = & 0\,1\,1\,0 \\ 3 = & \underline{0\,0\,1\,1} \\ = & \underline{1\,0\,0\,1} \end{array}$$

The transmitted codeword is thus:

11	10	9	8	7	6	5	4	3	2	1
1	0	0	*1*	1	1	0	*0*	1	*0*	*1*

Similarly, at the receiver, the 4-bit binary numbers corresponding to those bit positions with a binary 1, including the check bits, are again added together. If no errors have occurred, the modulo-2 sum is zero:

$$\begin{array}{rl} 11 = & 1\,0\,1\,1 \\ 8 = & 1\,0\,0\,0 \\ 7 = & 0\,1\,1\,1 \\ 6 = & 0\,1\,1\,0 \\ 3 = & 0\,0\,1\,1 \\ 1 = & \underline{0\,0\,0\,1} \\ = & \underline{0\,0\,0\,0} \end{array}$$

Now consider a single-bit error: say bit 11 is corrupted from 1 to 0. The new modulo-2 sum is now:

$$\begin{array}{rl} 8 = & 1\,0\,0\,0 \\ 7 = & 0\,1\,1\,1 \\ 6 = & 0\,1\,1\,0 \\ 3 = & 0\,0\,1\,1 \\ 1 = & \underline{0\,0\,0\,1} \\ = & \underline{1\,0\,1\,1} \end{array}$$

Firstly, the sum is nonzero, which indicates an error, and secondly, the modulo-2 sum, equivalent to decimal 11, indicates that bit 11 is the erroneous bit. The latter is inverted to obtain the corrected codeword and hence data bits.

It can also be shown that if two bit errors occur, the modulo-2 sum is nonzero, thus indicating an error, but the positions of the errors cannot be determined from the sum. The Hamming single-bit code can correct for single-bit errors and detect two-bit errors but other multiple-bit errors cannot be detected.

As we saw in Chapter 6, the main types of error in many data communication networks are error bursts rather than, say, isolated single or double-bit errors. Hence, although the Hamming coding scheme in its basic form appears to be inappropriate for use with such networks, a simple technique called interleaving is often used to extend the application of such a scheme.

Consider, for example, a requirement to transmit a block of data, comprising a string of, say, eight ASCII characters, over a simplex channel that has a high probability of an error burst of, say, seven bits. The controlling device first converts each ASCII character into its 11-bit codeword form to give a block of eight 11-bit codewords. Then, instead of transmitting each codeword separately, the controlling device transmits the contents of the block of codewords a column at a time. Thus the eight, say, most significant bits are transmitted first, then the eight next most significant bits and so on, finishing with the eight least significant bits. The controlling device at the receiver then performs the reverse operation, reassembling the transmitted block in memory, prior to performing the detection and, if necessary, correction operation on each codeword.

The effect of this approach is that if an error burst of up to seven bits does occur, it affects only a single bit in each codeword rather than a string of bits in one or two codewords. This means that, assuming just a single error burst in the 88 bits transmitted, the receiver can determine a correct copy of the transmitted block of characters.

Although the approach just outlined provides a way of extending the usefulness of this type of encoding scheme, Hamming codes are used mainly in applications that have isolated single-bit errors; an example is in error-correcting semiconductor memory systems. As we showed in Figure 11.18, the preferred method of achieving FEC in most digital communication systems is based on a combination of a Reed–Solomon block code and a convolutional coder. We shall now briefly describe the operation of convolutional coders.

B.3 Convolutional codes

Block codes are *memoryless* codes as each output codeword depends only on the current *k*-bit message block being encoded. In contrast, with a convolutional code, the continuous stream of source bits is operated upon to produce a continuous stream of output (encoded) bits. Because of the nature of the encoding process, the sequence of source bits is said to be convolved (by applying a specific binary operation on them) to produce the output bit sequence. Also, each bit in the output sequence is dependent not only on the current bit being encoded but also on the previous sequence of source bits, thus implying some form of memory. In practice, as we shall see, this takes the form of a shift register of a finite length, known as the **constraint length**, and the convolution (binary) operation is performed using one or more modulo-2 adders (exclusive-OR gates).

Encoding

An example of a convolutional encoder is shown in Figure B.1(a). With this encoder, the three-bit shift register provides the memory and the two modulo-2 adders the convolution operation. For each bit in the input sequence, two bits are output, one from each of the two modulo-2 adders. The encoder shown is thus known as a rate $1/2(k/n)$ convolutional encoder with a constraint length of 3.

Because of the memory associated with a convolutional encoder, we must have a convenient means of determining the specific output bit sequence generated for a given input sequence. Three techniques can be used, each based on a form of diagrammatic representation: a tree diagram, a state diagram, and a trellis diagram. In practice, the last is the most frequently used method because it is the most useful for demonstrating the decoding operation. However, before we can draw this, we must determine the outputs for each possible input sequence using either the tree or state diagram.

As an example, Figure B.1(b) shows the **tree diagram** for the encoder in Figure B.1(a). The branching points in the tree are known as nodes and the tree shows the two possible branches at each node; the upper of the two branches corresponds to a 0 input bit and the lower branch to a 1 bit. The pair of output bits corresponding to the two possible branches at each node are shown on the outside of each branch line.

As we can see, with a tree diagram the number of branches in the tree doubles for each new input bit. However, the tree is repetitive after the second branch level since, after this level, there are only four unique branch nodes. These are known as *states* and are shown as *A*, *B*, *C*, and *D* in the figure.

As we can see, from any one of these nodes the same pair of output bits and new node state occurs, irrespective of the position of the node in the tree. For example, from any node *C* the same pair of branch alternatives occur: 10 output and new state *A* for a 0 input, or 01 output and new state *B* for a 1 input.

Once we have identified the states for the encoder using the tree diagram, we can draw the **trellis diagram**. As an example, the trellis diagram for the same encoder is shown in Figure B.2(b). As we can see, after the second branch level, the repetitive nature of the tree diagram is exploited by representing all the possible encoder outputs in a more reduced form.

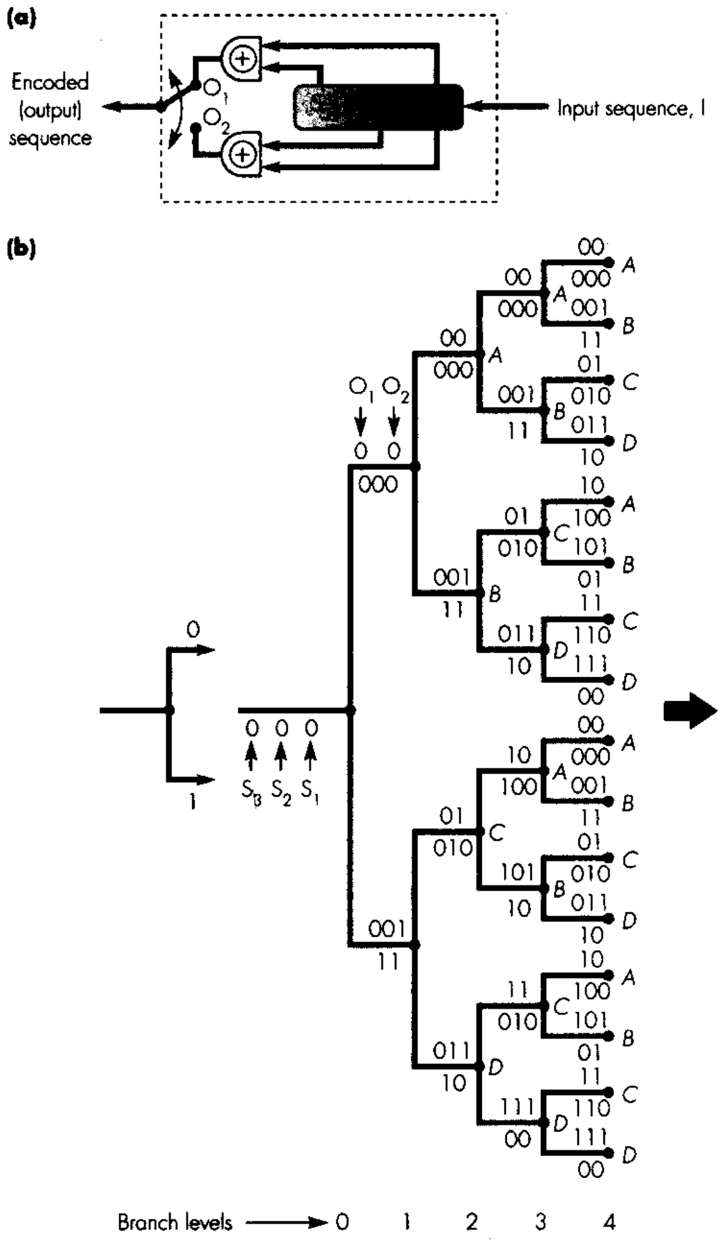
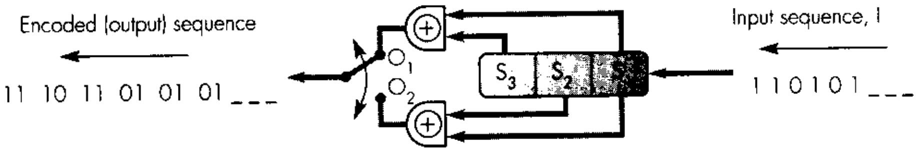


Figure B.1 Convolutional coder principles: (a) example encoder circuit; (b) tree diagram representaton of the encoder.

(a)



Input sequence I	Shift register contents S_1	 S_2	 S_3	Output sequence O_1	 O_2
1	1	0	0	1	1
1	1	1	0	1	0
0	0	1	1	1	1
1	1	0	1	0	1
0	0	1	0	0	1
1	1	0	1	0	1
⋮		⋮		⋮	

(b)

(c)

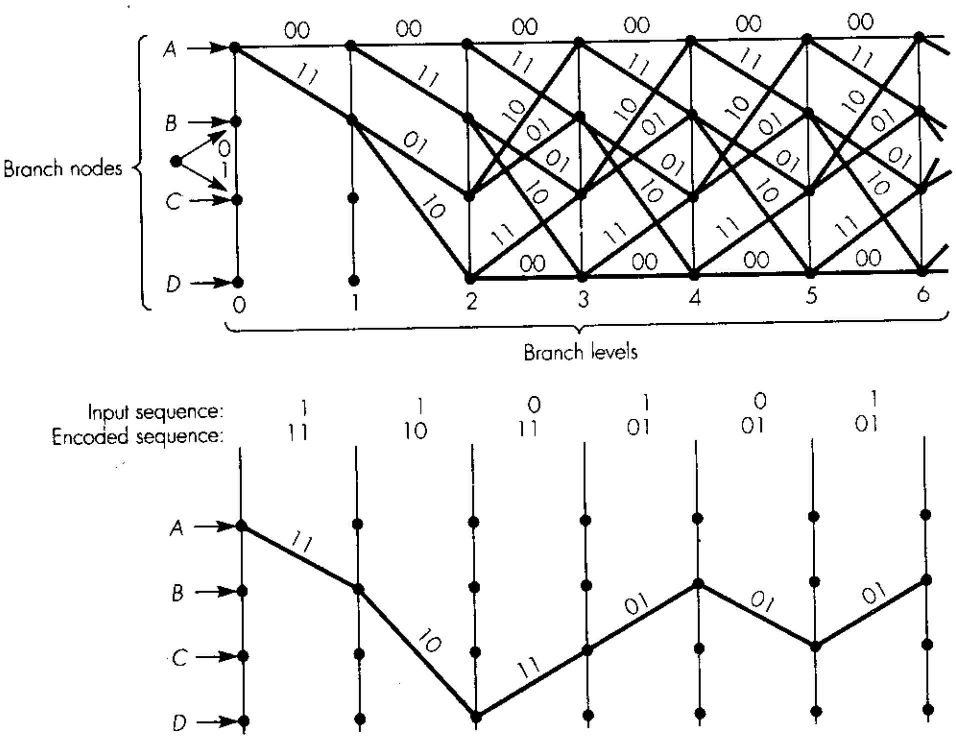


Figure B.2 Convolutional encoder output derivation: (a) circuit; (b) trellis diagram; (c) example output.

The trellis diagram shows the outputs that result from this encoder for all possible input bit sequences. Then, for a specific input sequence, a single path through the trellis – and hence sequence of output bits – results. As an example, Figure B.2(c) shows the path through the trellis, and hence the output sequence, corresponding to the input sequence 110101... .

Initially, we assume that the shift register is cleared, that is, it is set to all 0s. After the first bit in the input sequence has been shifted (entered) into the shift register, its contents are 001. The outputs from the two modulo-2 adders are 0 + 1 = 1 (adder 1) and 0 + 1 = 1 (adder 2). Thus, the first two output bits are 11 and these are output before the next input bit is entered into the shift register. Since the input bit was a 1, the lower branch path on the trellis diagram is followed and the output is 11, as derived.

After the second input bit has been entered, the shift register contains 011. The two adder outputs are 0 + 1 = 1 (adder 1) and 1 + 1 = 0 (adder 2). Thus, the two output bits are 10 and again these are output before the next input bit is processed. Again, since the input bit was a 1, the lower branch on the trellis diagram is followed and the output is 10, as derived. Continuing, the third input bit makes the shift register contents 110 and hence the two output bits are 11; 1 + 0 = 1 (adder 1) and 1 + 0 = 1 (adder 2). Also, since the input bit was a 0, the upper branch path on the trellis diagram is followed. This process then continues.

Decoding

The aim of the decoder is to determine the *most likely* output sequence, given a received bitstream (which may have errors) and a knowledge of the encoder used at the source. The decoding procedure is equivalent to comparing the received sequence with all the possible sequences that may be obtained with the respective encoder and then selecting the sequence that is closest to the received sequence. Recall that the Hamming distance between two codewords is the number of bits that differ between them. Therefore, when selecting the sequence that is closest to the received sequence, the Hamming distance between the received sequence and each of the possible sequences is computed, and the one with the least distance is selected. Clearly, in the limit this necessitates comparing the complete received sequence with all the possible sequences, and hence paths through the trellis. This is impractical in most cases and hence we must compromise.

Essentially, a running count is maintained of the distance between the actual received sequence and each possible sequence but, at each node in the trellis, only a single path is retained. There are always two paths merging at each node and the path selected is the one with the minimum Hamming distance, the other is simply terminated. The retained paths are known as **survivor paths** and the final path selected is the one with a continuous path through the trellis with a minimum aggregate Hamming distance. This procedure is known as the **Viterbi algorithm**. The decoder, which aims to find the most likely path corresponding to the received sequence, is known as a **maximum-likelihood decoder**. Example B.1 describes the Viterbi algorithm.

Example B.1

Assume that a message sequence of 1001110... is to be sent using the encoder shown in Figure B.1(a). From the trellis diagram for this encoder, we can deduce that this will yield a transmitted (output) sequence of:

11 01 10 11 10 00 11 ...

Now assume a burst error occurs so that two bits of this encoded sequence are corrupted during transmission. The received sequence is as follows:

11 01 00 11 11 00 11 ...
↑ ↑

Use the Viterbi algorithm to determine from this the most likely transmitted sequence.

Answer:

The various steps associated with the encoding and decoding procedures are shown in Figure B.3. Part (a) shows the path through the trellis corresponding to the original output from the encoder and part (b) shows how the survivor paths are chosen. The number shown by each path merging at a node in Figure B.3(b) is the accumulated Hamming distance between the path followed to get to that node and the actual received sequence.

If the path chosen is that starting at the root node (branch level 0), the received sequence is 11 and the Hamming distances for the two paths are 2 for path 00 and 0 for path 11. These two distance values are added to the paths emanating from these nodes. Thus, at branch level 1, the received sequence is 01 and the two paths from node *A* have Hamming distances of 1 for path 00 and 1 for path 11. The accumulated distances are thus 2 + 1 = 3 for each path. Similarly, the two paths emanating from node *B* have Hamming distances of 0 for path 01 and 2 for path 10, and hence the accumulated distances are 0 + 0 = 0 and 0 + 2 = 2, respectively. A similar procedure is repeated at branch level 2.

At branch level 3 and onwards, however, the selection process starts. Thus, the two paths merging at node *A* (at branch level 3) have accumulated distances of 3 and 1, of which the latter is selected to be the survivor path for this node – this is shown as a bold line on the trellis diagram. A similar selection process is followed at nodes *B*, *C*, and *D*. At node *C*, however, we can see that the two merging paths both have the same accumulated distance of 4. In such cases, the upper path is selected. Also, after the selection process, all subsequent distances are calculated relative to the accumulated distance associated with the selected path.

▶

B.1 Continued

[illegible] the output [illegible] by inspection [illegible]

- [illegible] trellis;
- [illegible] the minimum.

[illegible] output [illegible] to the origi- [illegible]

Finally, note that no FEC method can identify all errors. In general, codes like the convolutional code are used primarily to reduce the error probability (bit error rate) of a link to a more acceptable level. A typical reduction with a rate 1/2 convolutional coder is between 10^2 and 10^3. Hence when used with a Reed–Solomon (block) coder, the number of residual bit errors after the block decoding process is reduced to a level that is acceptable for most entertainment applications involving audio and video streams.

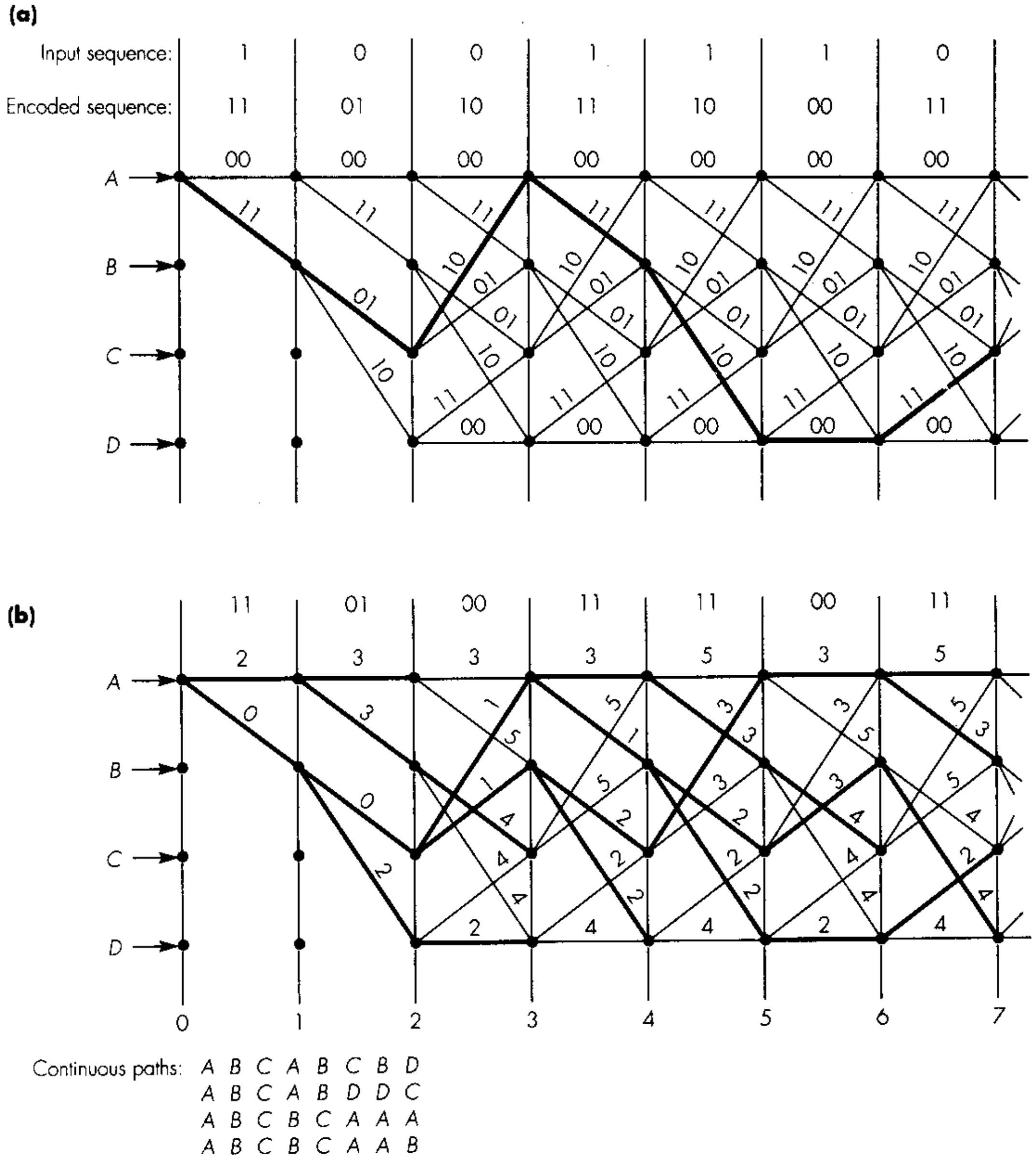


Figure B.3 Convolutional decoder output derivation: (a) actual decoder output; (b) derived survivor paths.

bibliography and further reading

In order to keep abreast of developments in the communications and networking fields, you should try to read on a regular basis the journals and magazines published in these fields by the major institutions. The most widely read journals that contain a range of technical papers in the area of communications and networks are: *IEEE Transactions on Communications, IEEE Journal on Selected Areas in Communications, IEE Proceedings on Communications, Communications of the ACM,* and *Computer Networks and ISDN.* In addition, the IEEE publish four magazines: *IEEE Network, IEEE Communications, IEEE Multimedia,* and *IEEE Internet Computing* all of which contain articles relating to state-of-the-art developments in their related fields.

The listing of titles within this Bibliography reflects the order in which the related topic is discussed in the text.

Chapter 1

Bell T.E. (1996). Communications. *IEEE Spectrum,* January

Decina M. and Trecordi V. (1999). Voice over IP and e-commerce. *IEEE Communications,* September

Comerford R. and Perry T.S. (1996). Wired for interactivity. *IEEE Spectrum,* April

IEEE (1998). *IEEE Network: Special Issue on Transmission and Distribution of Digital Video,* November

Riezenman M.J. (2000). Communications. *IEEE Spectrum,* January

Landon K.C. and Landon J.P. (1998). *Information Systems and the Internet.* Dryden Press

Comerford R. (2000). The Internet. *IEEE Spectrum,* January

Kwok T. (1995). A vision for residential broadband services. *IEEE Network,* September

Berners-Lee T. et al. (1994). The World Wide Web. *Communications of the ACM,* August

Chapter 2

Bellamy J. (1991). *Digital Telephony* 2nd edn. Wiley

Alber A.F. (1985). *Videotex/Teletext, Principles and Practices.* McGraw-Hill

Fisher S. (1995). *Multimedia Authoring.* Academic Press

Musciano C. and Kennedy B. (1996). *HTML, The Definitive Guide.* O'Reilly and Associates

Petersen D.(1992). *Audio, Video and Data Telecommunications.* McGraw-Hill

Nielsen J. (1995). *Multimedia and Hypertext.* AP Professional

Chapter 3

Hill R. (1986). *A First Course in Coding Theory.* Oxford: Clarendon Press

Shannon C. (1948). A mathematical theory of communication. *Bell System Tech. Journal*

Storer J.A. (1988). *Data Compression: methods and theory.* Computer Science Press

Vitter J.S. (1987). Design and analysis of dynamic Huffman codes. *Journal of the ACM,* October

Welch T.A. (1984). A technique for high performance data compression. *IEEE Computer*, June

Witten I.H. (1987). Arithmetic coding for data compression. *Comm. ACM*, June

ISO (1994). *Digital Compression and Coding of Continuous Tone Still Images (JPEG)*. (ISO 10918-1)

Wallace G.K. (1991). The JPEG still picture compression standard. *Comm. ACM*, September

Clarke R.J. (1995). *Digital compression of still images and video*. Academic Press

Chapter 4

Bellamy J. (1991). *Digital telephony* 2nd edn. Wiley

Cox R.V. and Kroon P. (1996). Low bit rate speech coders for multimedia communication. *IEEE Communications*, December

Ambikairajah E. et al. (1997). Auditory masking and MPEG-1 audio compression. *IEE Electronic and Communication Engineering*, August

Jayant N. et al. (1993). Signal compression based on models of human perception. *Proc. IEEE*, October

Pan D. (1995). A tutorial on MPEG audio compression. IEEE *Multimedia*, June

Sikora T. (1997). MPEG digital video coding standards. *IEEE Signal Processing*, September

Bhaskaran V. and Konstantinides K. (1995). *Image and video compression standards, algorithms and architectures*. Academic Press

Tudor P.N. (1995). MPEG-2 video compression. *IEE Electronic and Communication Engineering*, December

Ghanbari M. (1999) Video Coding: an introduction to standard codecs. *IEE Telecommunications*, Series 42

ITU-T(1990). *Video Codec for Audio Visual Services at p × 64 kbps*. Recommendation H.261

Rijkse K. (1996). H.263: Video coding for low bit rate communication. *IEEE Communications*, December

Chen C.T. and Wong A.(1993). A self-governing rate buffer control strategy for pseudoconstant bit rate video coding. *IEEE Trans. on Communications*, January

Redmill D.W. (1998). Error-resilient image and video coding. *IEE Electronics and Communication Engineering*, August

Benoit H. (1997). *Digital Television: MPEG-1, MPEG-2 and DVB*, Arnold

Koenen R. (1999). MPEG-4: Multimedia for our time. *IEEE Spectrum*, February

Schafer R. (1998). MPEG-4: A multimedia compression standard for interactive applications. *IEEE Electronics and Communication Engineering*, December

IEEE (1998). *IEE Trans. on Circuits and Systems: Special Issue on Segmentation for Video*, September

Pancha P. and El Zarki M. (1994). MPEG coding for variable bit rate transmission. *IEEE Communications*, May

Solari S.J. (1997). *Digital Video and Audio Compression*. McGraw-Hill

IEEE (1998). *IEEE Communications: Special Issue on Wireless Video*, June

Chapter 5

ITU-T (1995). *Narrowband ISDN Video Telephone Systems and Terminal Equipment*. (Recommendation H.320)

ITU-T (1995). *Terminal for Low Bit Rate Multimedia Communications*. (Recommendation H.324)

Lindbergh D. (1996). The H.324 multimedia communication standard. *IEEE Communications*, December

ITU-T (1995). *Adaptation of H.320 Terminals to B-ISDN Environments.* (Recommendation H.321)

ITU-T (1996). *Broadband Audio-visual Communications Systems and Terminals.* (Recommendation H.310)

ITU-T (1995). *Visual Telephone Systems and Terminals for LANs with Guaranteed QoS.* (Recommendation H.322)

ITU-T (1996). *Visual Telephone Systems and Terminals for LANs with Non-guaranteed QoS.* (Recommendation H.325)

Thom G.A. (1996). H.323: The multimedia communication standard for LANs. *IEEE Communications*, December

ITU-T (1996). *Media Stream Packetization and Synchronization (for H.323).* (Recommendation H.225)

ITU-T (1988). *PCM of Voice Frequencies.* (Recommendation G.711)

ITU-T (1988). *7 kHz Audio Coding within 64 kbps.* (Recommendation G.722)

ITU-T (1995). *Dual-rate Speech Codec at 6.4 and 5.3 kbps.* (Recommendation G.723.1)

ITU-T (1992). *Speech Coding at 16 kbps.* (Recommendation G.728)

ITU-T (1995). *Speech Codec for Multimedia Communications at 8/13 kbps.* (Recommendation G.729)

ITU-T (1993). *Video Codec for Audio-visual Communications at p × 64 kbps.* (Recommendation H.261)

ITU-T (1995). *Video Codec for Low Bit Rate Channels less than 64 kbps.* (Recommendation H.263)

ITU-T (1994). *Transmission Protocols for Multimedia Data.* (Recommendation T.120)

ITU-T standards are available at *http://www.itu.ch*

Postel J. (1981). *The Internet Protocol (IP)*, September (RFC 791)

Postel J. (1981). *The Transmission Control Protocol (TCP)*, September (RFC 793)

Postel J. (1982). *The Simple Mail Transfer Protocol (SMTP)*, August (RFC 821)

Crocker D.H.(1982). *Standard Format for Internet Text Messages*, August (RFC 822)

Mockapetris P.V. (1987). *Domain Names: Concepts and Facilities*, November (RFC 1034)

Myers J. and Rose M. (1996). *Post Office Protocol, Version 3 (POP3)*, May (RFC 1939)

Freed N. et al. (1996). *Multipurpose Internet Mail Extensions (MIME)*, November (RFC 2045/6/7/8)

Berners-Lee T. et al. (1996). *The Hypertext Transfer Protocol. HTTP/1.0*, May (RFC 1945)

Postel J. and Reynolds J. (1985). *The File Transfer Protocol (FTP)*, October (RFC 959)

Kantor B. and Lapsley P. (1986). *The Network News Transfer Protocol (NNTP)*, February (RFC 977)

RFC standards information is available at *http://info.internet.isi.edu/in-notes/rfc/files/rfc xyz.txt*

ISO/IEC (1991). *MPEG-1: Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to 1.5 Mbps.* (ISO/IEC Recommendation 11172)

ISO/IEC (1994). *MPEG-2: Coding of Moving Pictures and Associated Audio Information.* (ISO/IEC Recommendation 13818-2)

Chapter 6

Halsall F. (1996). *Data Communications, Computer Networks and Open Systems.* Addison-Wesley

Spragins J.D. et al. (1991). *Telecommunications Protocols and Design.* Addison-Wesley

Black U.D. (1993). *Data Link Protocols.* Prentice Hall

Marano K. et al. (1990) Echo cancellation and applications. *IEEE Communications.* January

Pearson J.E. (1992). *Basic Communication Theory.* Prentice Hall

Sklar B. (1988). *Digital Communications: fundamentals and applications.* Prentice Hall

Fletcher J. (1982). An arithmetic checksum for serial transmissions. *IEEE Trans. on Communications,* January

Ramabadroan T. (1988). A tutorial on CRC computations. *IEEE Micro,* August

Choi T.Y. (1985). Formal techniques for the specification of protocols. *IEEE Communications,* January

Conrad J. (1980). Character-oriented data link control protocols. *IEEE Trans. on Communications,* April

Danthine A.A.S. (1970). Protocol representation with finite-state models. *IEEE Trans. on Communication,* April.

Pouzin L. and Zimmerman H. (1978). A tutorial on protocols. *Proc. IEEE,* November

Carlson D.E (1980). Bit-oriented data link control procedures. *IEEE Trans. on Communications,* April

IEEE (1985). *Logical Link Control (LLC).* (IEEE Standard 802.2)

Chapter 7

Bellamy J.(1991). *Digital Telephony* 2nd edn. Wiley

Flood J.E. (1995). *Telecommunications Switching, Traffic and Networks.* Prentice Hall

Ericsson (1987). *Telecommunications: telephone networks 1 and 2.* Chartwell-Bratt

Nortel (1992). *Synchronous Transmission Systems.* Northern Telecom Europe

Omidyar C.G. and Alridge A. (1993). Introduction to SDH/SONET. *IEEE Communications,* September

Fergusson S.P. (1994). Implications of SONET and SDH. *IEE Electronic and Communication Engineering,* June

Flood J. and Cochraine P. (1991). *Transmission Systems.* Peter Peregrinus

Bissell C.C. and Chapman D.A. (1992). *Digital Signals Transmission.* Cambridge University Press

Freeman R.L.(1989). *Telecommunication System Engineering* 2nd edn. Wiley

Stallings W. (1989). *ISDN: an introduction.* Macmillan

Held G. (1994). *The Complete Modem Reference* 2nd edn. Wiley

EIA (1987). *Standard Interface between a DTE and DCE for Serial Data Interchange.* (EIA-232D)

Lechleider J.W. (1989). Line codes for digital subscriber lines. *IEEE Communications,* September

Doyle J.S. and McMahon C.S. (1988). The intelligent network concept. *IEEE Trans. on Communications,* December

Chapter 8

Metcalf R.M. and Boggs D.R. (1976). Ethernet: distributed packet switching for local computer networks. *Comm. ACM,* July

IEEE (1985). *802.3 CSMA/CD Access Method and Physical Layer Specifications*

Bux W. et al. (1983). Architecture and design of a reliable token-ring network. *IEEE Journal on Selected Areas in Communications*, November

IEEE (1985). *802.5 Token Ring Access Method and Physical Layer Specifications*

Stuck B.W. (1983). Calculating the maximum throughput rate of LANs. *IEEE Computer*, May

Backes F. (1988). Transparent bridges for interconnection of IEEE 802 LANs. *IEEE Network*, January

Bederman S. (1986). Source routing. *Data Communications*, February

Dixon R. and Pitt D. (1988). Addressing, bridging and source routing. *IEEE Network*, January

Perlman R. (1992). *Interconnections: bridges and routers*. Addison-Wesley

IEEE (1988). *802.1D MAC Bridges*

Hart J. (1988). Extending the IEEE 802.1 bridge standard to remote bridges. *IEEE Network*, January

IEEE (1985). *802.5 (Appendix D) Multiring Networks – Source Routing*

Ross F.E. (1986). FDDI – A tutorial. *IEEE Communications*, May

Ross F.E. (1989). An overview of FDDI. *IEEE Journal on Selected Areas of Communications*, September

Jain R. (1994). *FDDI Handbook: high-speed networking using fiber and other media*. Addison-Wesley

Molle M. and Watson G. (1996). 100Base T and IEEE 802.12 packet switching. *IEEE Communications*, August

Ozveren C. (1994). Reliable and efficient hop-by-hop flow control. *Sigcomm 94*

IEEE (1995). *802.3 u MAC and Physical Layer Parameters for 100Base T*

Cisco (1998). *Designing Switched LAN Internetworks*. Cisco Systems Inc.

Cisco (1999). *LAN Switches*. Cisco Systems Inc.

Molle M. (1997). Frame bursting: a technique for scaling CSMA/CD to gigabit speeds. *IEEE Network*, July

Frazier H. and Johnson H. (1999). Gigabit Ethernet: from 100 to 1000 Mbps. *IEEE Internet Computing*, January

Haddock S. (1996). *Carrier Extension Issues*. IEEE 802.3 Study Group Meeting, Netherlands, July

IEEE (1985). *Logical Link Control (LLC) Protocol*, (Standard 802.2)

Duc N. and Chew E. (1985). ISDN protocol architecture. *IEEE Communications*, March

Bush J. (1989). Frame relay promises WAN bandwidth on demand. *Data Communications*, July

Lai W. (1989). Frame relaying service: an overview. *IEEE InfoCom 89*

Smith P. (1993). *Frame Relay*. Addison-Wesley

Chapter 9

Comer D.E. (1995). *Internetworking with TCP/IP* 3rd edn. Prentice Hall

Postel J. (1980). *The Internet Protocol*, September (RFC 791)

Stevens W.R. (1994). *TCP/IP Illustrated* Vol.1. Addison-Wesley

Huitema C. (1995). *Routing in the Internet*. Prentice Hall

Sahasrabuddhe L.H. and Mukherjee B. (2000). Multicast routing: a tutorial. *IEEE Network*, January

Mills D. (1984). *The EGP Formal Specification*, April (RFC 904)

Braden R.T. et al. (1988). *Computing the Checksum for the IP Protocols*, September (RFC 1071)

Plummer D.C. (1982). *An Ethernet Address Resolution Protocol*. November (RFC 826)

Finlayson R. et al. (1984). *A Reverse Address Resolution Protocol,* June (RFC 903)

Hendrick C.L. (1988). *Routing Information Protocol,* June (RFC 1058)

Waitzman D. et al. (1988). *Distance Vector Multicast Routing Protocol,* November (RFC 1075)

Deering S. (1991). *ICMP Router Discovery Messages,* September (RFC 1256)

Moy J. (1991). *OSPF Version 2,* July (RFC 1247)

Moy J. (1994). *Multicast Extensions to OSPF,* March (RFC 1584)

Simpson W. (1994). *The Point-to-Point Protocol (PPP),* July (RFC 1661)

Simpson W. (1994). *PPP in HDLC-like Framing,* July (RFC 1662)

Metz C. (1999). A pointed look at the PPP. *IEEE Internet Computing,* July

Xiao X. and Ni L.M. (1999). Internet QoS: a big picture. *IEEE Network,* March

Metz C. (1999). RSVP: general purpose signaling for IP. *IEEE Internet Computing,* May

Braden R. (1997). *RSVP Version 1 Functional Specification,* September (RFC 2205)

Blake S. et al. (1998). *An Architecture for Differentiated Services,* December (RFC 2475)

Bernet Y. (2000). The complementary roles of RSVP and differentiated services in the full-service QoS network. *IEEE Communications,* February

Wang B. and Hou J.C (2000). Multicast-routing and its QoS extension. *IEEE Network,* January

Armitage G. (2000). MPLS: The magic behind the myths, *IEEE Communications,* January

Newman P. et al. (1997). IP switching and gigabit routers. *IEEE Communications,* January

Deering S. and Hinden R. (1995) *IPv6 Specification,* December (RFC 1883)

Hinden R. and Deering S. (1995). *IPv6 Addressing Architecture,* December (RFC 1884)

Floyd S. and Jacobson V. (1993). Random early detection gateways for congestion avoidance. *IEEE Trans. on Networking,* August

Gilligan R. and Nordmark E. (1996). *Transition: Mechanisms for IPv6 Hosts and Routers,* April (RFC 1993)

Huitema C. (1997). *IPv6: The New Internet Protocol.* Prentice Hall

Chapter 10

LeBoudec J.L. (1992). The asynchronous transfer mode: a tutorial. *Computer Networks and ISDN*

ATM Forum (1994). *ATM User/Network Interface Specification,* July

Lee B.G. et al. (1996). *Broadband Telecommunications Technology* 2nd edn. Artech House

Cuthbert L.G. (1993). *ATM: the broadband telecommunications solution.* IEE London

Stallings W. (1992). *ISDN and Broadband ISDN* 2nd edn. Macmillan

Händel R. et al. (1994). *ATM Networks: concepts, protocols, applications.* Addison-Wesley

Heinanen J. (1993). *Multiprotocol Encapsulation over ATM Adaption Layer 5.* (RFC 1483)

ITU-T (1993). *B-ISDN Adaptation Layer (AAL) Specification.* (Recommendation I.363)

ITU-T (1993). *Support of Broadband Connectionless Data Service on B-ISDN.* (Recommendation I.364)

Rao S.K. and Hatamian M. (1995). The ATM physical layer. *Complete Comms. Review,* April

Karak N. (1995). Data communications in ATM networks. *IEEE Network,* May

Lanbach M. (1994). Classical IP and ARP over ATM. (RFC 1577)

McDysan D.E. and Spohn D.L.(1995). *ATM Theory and Application.* McGraw-Hill

Newman P. (1994). ATM local area networks. *IEEE Communications,* March

Partridge C. (1994). *Gigabit Networking.* Addison-Wesley

Chao H.J. et al. (1994). IP on ATM local area networks. *IEEE Communications*, August

Truong H.L. et al. (1995). LAN emulation on an ATM network. *IEEE Communications*, May

Finn N. and Mason T. (1996). ATM LAN emulation. *IEEE Communications*, June

Dixit S. and Elby S. (1996). Frame relay and ATM interworking. *IEEE Communications*, June

Akyildig I.F. and Bernhardt (1997). ATM LANs: a survey of requirements, architectures, and standards. *IEEE Communications*, July

Griffiths J.M. (1996). ATM: customer needs in the transition to B-ISDN. *IEE Electronic and Communication Engineering*, October

White P.P. (1998). ATM switching and IP routing integration. *IEEE Communications*, April

Luetchford J.C. et al (1998). Applications of ATM in global networks. *IEEE Communications*, August

Sadiku M.N.O. and Arvind A.S. (1994). Bibliography on DQDB. *Computer Comms. Review*, January

Guarene U. et al (1998). IP and ATM integration perspective. *IEEE Communications*, January

Gerla M. et al. (1993). Internetting LANs and MANs to B-ISDN for connectionless traffic support. *IEEE JSAC*, October

IEEE (1996) *IEEE Communications Special Issue: An introduction to flow and congestion control in ATM networks*, November

Chapter 11

IEEE (1997). *IEEE Network: Special Issue on Broadband Services over Residential Networks*, January

Bisdikian C. et al. (1996). Residential broadband data services over HFC networks. *IEEE Communications*, November

Pugh W. and Boyer G. (1995). Broadband access: comparing alternatives, *IEEE Communications*, August

Armbruster H. (1997). Information infrastructures and multimedia communications. *IEEE Communications*, September

Khasnabish B. (1997). Broadband to the home: architectures and access methods. *IEEE Network*, January

LinY-D. (1998). The IEEE 802.14 MAC protocol. *IEEE Communications Surveys*, October

Cable Television Laboratories Inc. (1999). *Data-over-cable Service Interface Specification: Radio Frequency Interface Specification*. http://www.cablemodem.com

Kiniry J.R. and Metz C. (1998). Cable modems: cable TV delivers the Internet. *IEEE Internet Computing*, May

Droms R. (1997). *Dynamic Host Configuration Protocol (DHCP)*, March (RFC 2131)

Paff A. (1995). Hybrid fiber/coax in the public telecommunications infrastructure. *IEEE Communications*, April

Benoit H. (1999). *Satellite Television: techniques of analogue and digital television*. Arnold

Arcidiacono A. (1997). Multimedia services and data broadcasting via satellite. *IEE Electronic and Communication Engineering Journal*, February

Drury G.M. (1997). DVB channel coding standards for broadcasting compressed video services. *IEE Electronic and Communication Engineering Journal*, February

Reimers U. (1997). DVB-T: the COFDM-based system for terrestrial broadcast television. *IEE Electronic and Communication Engineering Journal*, February

Yasuda H. and Ryan H.J.F. (1998). DAVIC and interactive multimedia services. *IEEE Communications*, September

Czajkowski I.K. (1999). High-speed copper access: a tutorial overview. *IEE Electronic and Communication Engineering Journal*, June

Maxwell K. (1996). ADSL: interim technology for the next 40 years. *IEEE Communications*, October

Saltzberg B.R. (1998). Comparison of single-carrier and multitone digital modulation for ADSL applications. *IEEE Communications*, November

Cioffi J.M. et al. (1999). Very-high-speed digital subscriber lines. *IEEE Communications*, April

IEEE (1999). *IEEE Communications Special issue on Broadband Access Copper Technologies*, May

Sweeney P. (1991). *Error Control Coding: an introduction*, Prentice Hall

Chapter 12

Socolofsky T. and Kale C. (1991). *A TCP/IP Tutorial*, January (RFC 1180)

Stevens W.R. (1994). *TCP/IP Illustrated* Vol.1. Addison-Wesley

Comer D.E. (1995). *Internetworking with TCP/IP* Vol.1, 3rd edn. Prentice Hall

Black U.D. (1995) *TCP/IP and Related Protocols*. McGraw-Hill

Stevens W.R. (1990). *UNIX Network Programming*. Prentice Hall

Postel J.B. (1981). *The Transmission Control Protocol (TCP)*, September (RFC 793)

Nagle J. (1984). *Congestion Control in TCP/IP Internetworks*, January (RFC 896)

Mogul J.C. (1993). IP network performance. In *Internet System Handbook*. Addison-Wesley

Karn P. and Partridge C. (1987). Improving RTT estimates in reliable transport protocols. *Computer Comms. Review*, August

Jacobson V. (1988). Congestion avoidance and control. *Computer Comms. Review*, August

Jacobson V. (1990b). Modified TCP congestion avoidance algorithm, *end2end mailing list*, April

Clark D.D. (1982). *Windows and Acknowledgement Strategy*, July (RFC 813)

Mogel J.C. and Deering S.E. (1990). *Path MTU Discovery*, April (RFC 1191)

Braden R.T. (1992a) *TIME-WAIT Association Hazards in TCP*, May (RFC 1337)

Jacobson V., Braden S. and Borman D. (1992). TCP Extensions for High Performance, May (RFC 1323)

Stevens W. (1997). *TCP Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery Algorithms*, January (RFC 2001)

Mathis M. et al. (1996). *TCP Selective Acknowledgement Options*, October (RFC 2018)

Allman M., Paxson V. and Stevens W. (1999). *TCP Congestion Control*, April (RFC 2581)

Postel J.B. (1980). *The User Datagram Protocol (UDP)*, August (RFC 768)

Partridge C. and Pink S. (1993). A faster UDP. *IEEE Trans. on Networking*, August

Schultzrinne H. et al. (1989). *RTP A Transport Protocol for Real-time Applications* – also includes specification of RTCP. (RFC 1889)

Brackmo L. and Peterson L. (1995). TCP Vegas: end-to-end congestion avoidance on a global Internet. *IEEE Journal on Selected Areas in Communications*, October

Chapter 13

Hoschka P. (1997). *ASN.1 homepage*, http://www-sop.inria.fr/rodeo/personnel/hoschka/asn1.html

ISO (1988). *ASN.1 and its Encoding Rules* (IS 8824/5)

Tanteprasut D. et al. (1997). ASN.1 Protocol specification for use with arbitrary encoding schemes. *IEEE Trans. on Networking*, August

Denning D. (1997). *Internet Besieged: countering cyberspace scofflaws.* Addison-Wesley

Diffie W. and Landau S. (1998). *Privacy On The Line: the politics of wire-tapping and encryption.* MIT Press

Kahn D. (1967). *The Code Breakers: the story of secret writing.* Macmillan

Netscape (1998). *Introduction to Public-Key Cryptography.* Netscape Communications Corp.

Kaufman C. et al. (1995). *Networking Security: private communication in a public world.* Prectice Hall

Kessler G.C. (1998). *An Overview of Cryptography,* Hill Associates, available on-line at http://www.hill.com/library/staffpubs/crypto.html

NIST (1993). Data Encryption Standard (DES). *National Institute of Standards and Technology, Processing Standards Publication 46-2*

NIST (1999b). Triple DES standard. *Draft Federal Information Processing Standard (FIPS) 46–3*

Kummert H. (1998). *The PPP Triple-DES Encryption Protocol,* September (RFC 2420)

Lai X. and Massey J. (1990). A proposal for a new block encryption standard. Advances in cryptology. *Proceedings of Eurocrypt 90.* Springer-Verlag

Rivest R.L., Shamir A. and Adelman L. (1978). On a method for obtaining digital signatures and public key cryptosystems. *Comm. ACM,* February

Kaliski B. and Staddon J. (1998). *PKCS#1-RSA Cryptography Specifications Version 2.* (RFC 2437)

Rivest R.L.(1992). *The MD5 Message-digest Algorithm,* April (RFC 1321)

Neuman B.C. and TS'O T. (1994). Kerberos: an authentication service for computer networks. *IEEE Communications,* September

Kohl J. and Neuman B.C. (1993). *The Kerberos Network Authentication Service,* V5, September (RFC 1510)

Maughan D. et al. (1998). *Internet Security Association and Key Management Protocol (ISAKMP), November* (RFC 2408)

IEEE (1997). *IEEE Network: Special Issue on Network and Internet Security,* May

IEEE (1999). *IEEE Network: Special Issue on Networking Security,* November

ITU-T (1993). *Directory: Authentication Framework,* November (Recommendation X.509)

Kent S. (1993). *Certificate-based Key Management,* February (RFC 1422)

Netscape (1999). *Netscape Certificate Server FAQ.* Netscape Communications Corporation

Chapter 14

Mockapetris P.V. (1983). *Domain Names: Implementation Specification,* November (RFC 833)

Mockapetris P.V. (1987). *Domain Names: Concepts and Facilities,* November(RFC 1034)

Mockapetris P.V. (1987). *Domain Names: Implementation and Specification,* November (RFC 1035)

IANA (1999). Internet Assigned Number Authorities, http://www.iana.org/

Albitz P. and Liu C. (1993). *DNS and BIND.* O'Reilly and Associates

Reynolds J. and Postel J. (1994). *Assigned Numbers,* October (RFC 1700)

Kille S. (1995). *A String Representation of Distinguished Names,* March (RFC 1779)

Vixie P. et al. (1997). *Dynamic Updates in the DNS,* April (RFC 2136)

Postel J.B. (1982). *Simple Mail Transfer Protocol (SMTP),* August (RFC 821)

Crocker D.H. (1982). *Standard Format for Internet Text Messages.* August (RFC 822)

Myers J. and Rose M. (1996). *Post Office Protocol, Version 3 (POP3)*, May (RFC 1939)

Freed N. and Borenstein N. (1996). *Multipurpose Internet Mail Extensions*, November (RFC 2045/8)

Zimmerman P.R. (1995). *The Official PGP User Guide.* MIT Press

Zimmerman P.R. (1995). *PGP: Source Code and Internals.* MIT Press

Rivest R.L. (1992). *The MD5 Message-digest Algorithm;* April (RFC 1321)

Postel J. and Reynolds J. (1985). *The File Transfer Protocol (FTP)*, October (RFC 959)

IEEE (1999). *IEEE Network: Special Issue on Internet Telephony*, May

Kostas T.J. et al. (1998). Real-time voice over packet-switched networks. *IEEE Network*, January

Morgan S. (1998). The Internet and the local telephone network: conflicts and opportunities. *IEEE Communications*, January

Dalgic I. et al. (1999). True number portability and advanced call screening in a SIP-based IP telephony system. *IEEE Internet Computing*, July

Asatani K. (1998). *Standardization on multimedia communications: computer-telephony-integration related issues. IEEE Communications*, July

Stallings W. (1993). *SNMP, SNMPv2, and CMIP.* Addison-Wesley

McCloghrie K. and Rose M.T. (1991). *MIB-II, for Network Management of TCP/IP Internets*, March (RFC 1213)

IEEE (1997). *IEEE Communications: Special Issue on Network Management*, October

IEEE (1999). *IEEE Network: Special Issue on Networking Security*, November

Chapter 15

Berners-Lee T. et al. (1994). The World-Wide Web. *Comm. ACM*, August

Vetter R.J. et al. (1994). Mosaic and the World-Wide Web. *IEEE Computer*, October

World Wide Web Consortium home page: http://www.w3.org

Khare R. (1999). Anatomy of a URL. *IEEE Internet Computing*, September

Berners-Lee T. et al.(1996). *Hypertext Transfer Protocol, HTTP/1.0*, May (RFC 1945)

Fielding R. et al. (1997). *Hypertext Transfer Protocol, HTTP 1.1*, January (RFC 2068)

Cheswick W.R. and Bellovin S.M.(1994): *Firewalls and Internet Security.* Addison-Wesley

Baentsch M. et al. (1997). Web caching: the application-level view of the Internet. *IEEE Communications*, June

Luotoren A. and Altis K. (1994). World Wide Web Proxies. *Computer Networks and ISDN*, November

Berghel H.L. (1996). The client side of the Web. *Comm. ACM*, January

Kwan T.T. et al. (1995). NCSA's WWW server: design and performance. *IEEE Computer*, November

Meyers P. (1998). *The HTML Web Classroom.* Prentice Hall

Bryan M. (1997). *SGML and HTML Explained.* Addison-Wesley

Rule J. (1999). *Dynamic HTML.* Addison-Wesley

Castro E. (1997). *Perl and CGI for the World Wide Web.* Addison-Wesley

Pancha P. and El Zarki M. (1994). MPEG coding for variable bit rate video transmission. *IEEE Communications*, May

Schulzrinne H. et al. (1998). *Real Time Streaming Protocol (RTSP)*, April (RFC 2326)

Real Networks. *RTSP Resource Center.* http://www.real.com/devzone/library/fireprot/rtsp/

Campione M. and Walrath K. (1996). *The Java Language Tutorial.* Addison-Wesley

Arnold K. and Gosling J. (1996). *The Java Programming Language.* Addison-Wesley

Van der Linden P. (1996). *Just Java.* Prentice Hall

Sun Microsystems. *Java Tutorial.* http://java.sun.com/nav/read/tutorial/index.html

Netscape. *Java Script.* http://home.netscape.com/eng/mozilla/2.0/handbook/javascript/

Dierks T. and Allen C. (1998). *The TLS Protocol,* January (RFC 2246)

MasterCard (1999). *SET Secure Electronic Transaction.* http://www.mastercard.com/shoponline/set/

Merkow M. et al. (1998). *Building SET Applications for Secure Transactions.* Wiley

IEEE (1999). *IEEE Communications: Special Issue on Electronic Commerce,* September

IEEE (1997). *IEEE Internet Computing: Special Issue on Electronic Commerce,* November

IEEE (1997). *IEEE Spectrum: Special Issue on Technology and the Electronic Economy,* February.

IEEE (1997). *IEEE Network: Special Issue on Network and Internet Security,* May

index

BOOK BANK COPY

Srinivas Institute of Technology
Acc No. 21095
Call No.