Data and Computer Communications

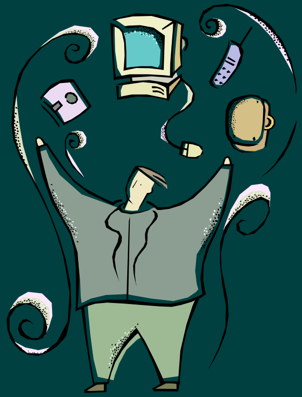
Tenth Edition

by William Stallings

CHAPTER 16

Advanced Data Communications Topics

"Life in the modern world is coming to depend more and more upon technical means of communication. Without such technical aids the modern city-state could not exist, for it is only by means of them that trade and business can proceed; that goods and services can be distributed where needed; that railways can run on schedule; that law and order are maintained; that education is possible. Communication renders true social life practicable, for communication means organization."



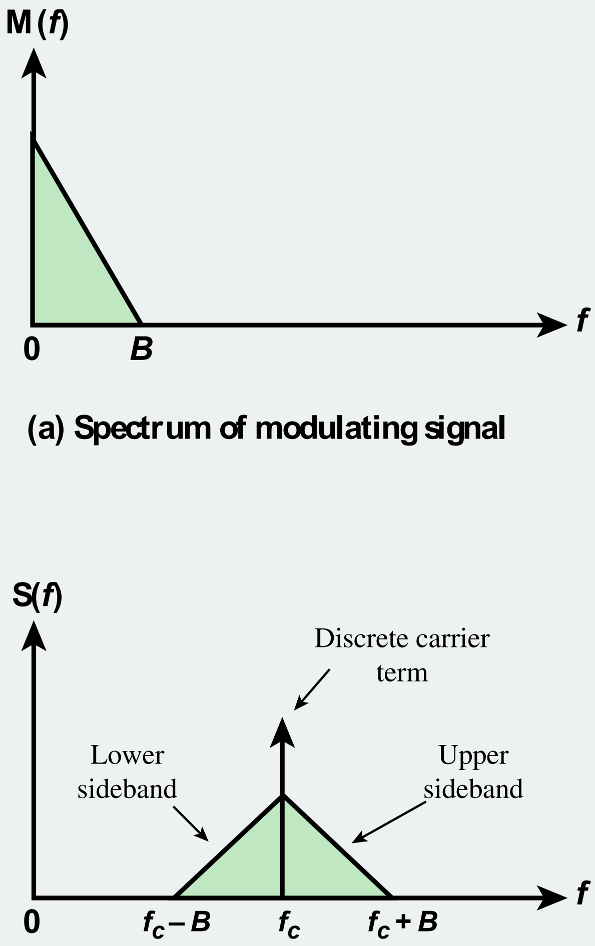
—On Human Communication,
Colin Cherry

Analog Data, Analog Signals

- There are two principal reasons for analog modulation of analog signals:
  - A higher frequency may be needed for effective transmission
  - Modulation permits frequency-division multiplexing

Principal techniques for modulation using analog data are:

- Amplitude modulation (AM)
- Frequency modulation (FM)
- Phase modulation (PM)



(a) Spectrum of modulating signal

(b) Spectrum of AM signal with carrier at f_c

Figure 16.2 Spectrum of an AM Signal

Angle Modulation

- Special cases are frequency modulation (FM) and phase modulation (PM)
- For phase modulation the phase is proportional to the modulating signal
- For frequency modulation the derivative of the phase is proportional to the modulating signal

Cyclic Codes

- Many of the error-correcting block codes that are in use are in this category
  - This class of codes can be easily encoded and decoded using linear feedback shift registers (LFSRs)
  - Bose-Chaudhuri-Hocquenghem (BCH) and Reed-Solomon are examples

- LFSR implementation is the same as that of the CRC (cyclic redundancy check) error-detecting code
  - Key difference is that the CRC code takes an input of arbitrary length and produces a fixed-length CRC check code while a cyclic error-correcting code takes a fixed-length input and produces a fixed-length check code

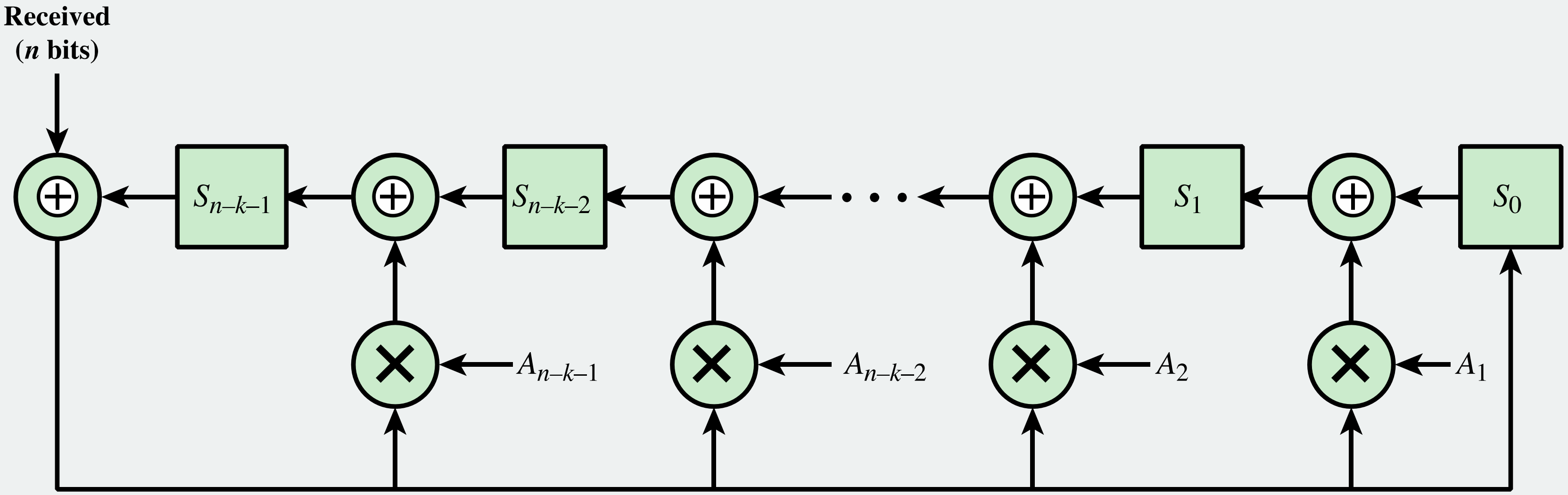
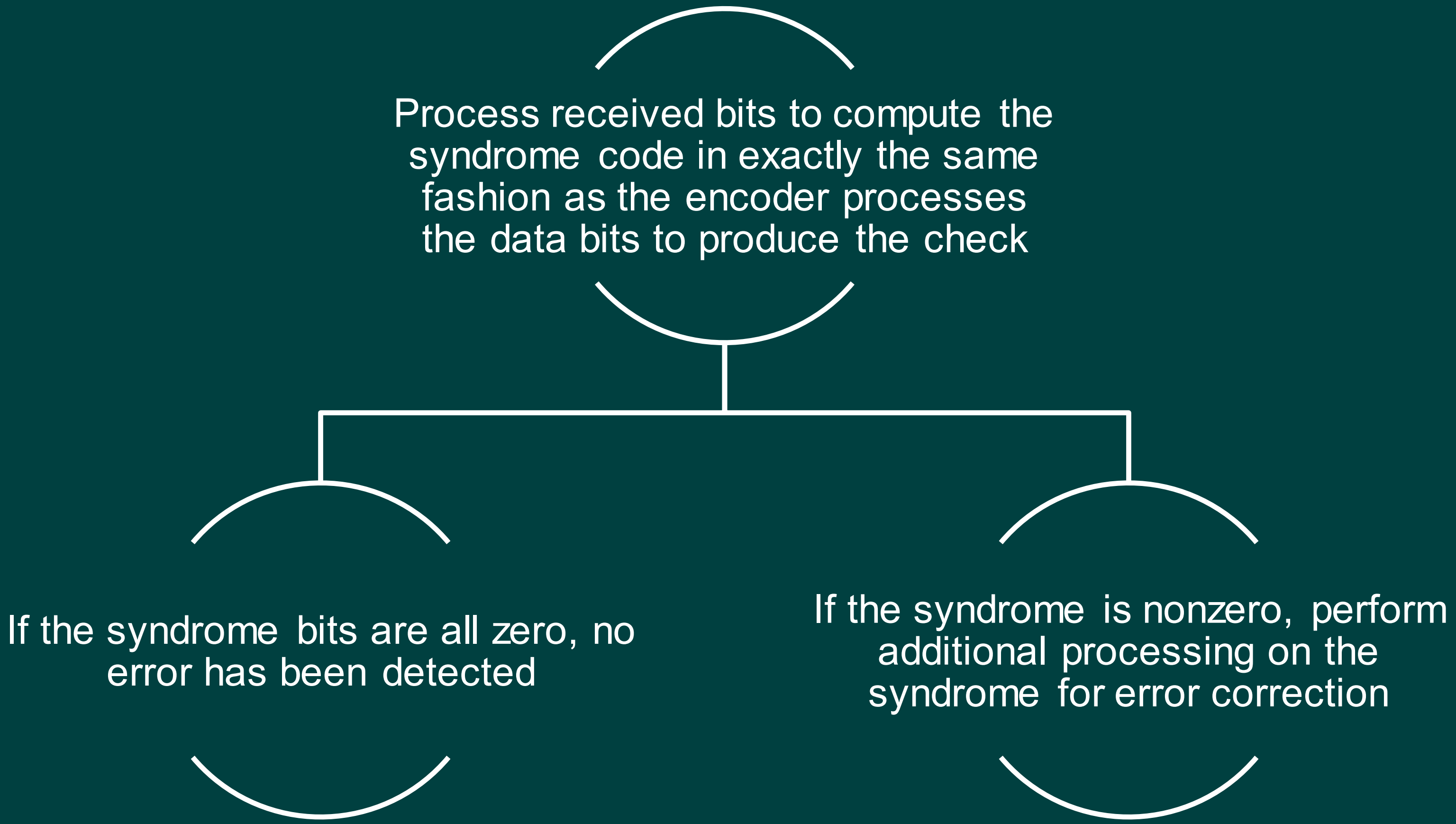


Figure 16.4 Block Syndrome Generator for Divisor
$(1 + A_1X + A_2X^2 + \ldots + A_{n-k-1}X^{n-k-1} + X^{n-k})$

Decoding of a Cyclic Code

- The following procedure is used:



Process received bits to compute the syndrome code in exactly the same fashion as the encoder processes the data bits to produce the check

If the syndrome bits are all zero, no error has been detected

If the syndrome is nonzero, perform additional processing on the syndrome for error correction

(a) Table of valid codewords	
Data Block	Codeword
0000	0000000
0001	0001101
0010	0010111
0011	0011010
0100	0100011
0101	0101110
0110	0110100
0111	0111001
1000	1000110
1001	1001011
1010	1010001
1011	1011100
1100	1100101
1101	1101000
1110	1110010
1111	1111111

(b) Table of syndromes for single-bit errors	
Error pattern E	Syndrome S
0000001	001
0000010	010
0000100	100
0001000	101
0010000	111
0100000	011
1000000	110

Table 16.1

A Single-Error-Correcting (7, 4) Cyclic Code

BCH Codes

- Among the most powerful cyclic block codes
- Widely used in wireless applications
- Provide flexibility in the choice of parameters (block length, code rate)
- For any positive pair of integers *m*and *t*, there is a binary *(n,k)* BCH code with the following parameters:
  - Block length: $n = 2^m - 1$
  - Number of check bits: $n - k \leq mt$
  - Minimum distance: $d_{\text{min}} \geq 2t + 1$

Table 16.2
BCH Code Parameters

n	*k*	*t*	*n*	*k*	*t*	*n*	*k*	*t*	*n*	*k*	*t*	*n*	*k*	*t*
7	4	1	63	30	6	127	64	10	255	207	6	255	99	23
15	11	1		24	7		57	11		199	7		91	25
	7	2		18	10		50	13		191	8		87	26
	5	3		16	11		43	14		187	9		79	27
31	26	1		10	13		36	15		179	10		71	29
	21	2		7	15		29	21		171	11		63	30
	16	3	127	120	1		22	23		163	12		55	31
	11	5		113	2		15	27		155	13		47	42
	6	7		106	3		8	31		147	14		45	43
63	57	1		99	4	255	247	1		139	15		37	45
	51	2		92	5		239	2		131	18		29	47
	45	3		85	6		231	3		123	19		21	55
	39	4		78	7		223	4		115	21		13	59
	36	5		71	9		215	5		107	22		9	63

Table 16.3
BCH Polynomial Generators

n	k	t	$P(X)$
7	4	1	$X^3 + X + 1$
15	11	1	$X^4 + X + 1$
15	7	2	$X^8 + X^7 + X^6 + X^4 + 1$
15	5	3	$X^{10} + X^8 + X^5 + X^4 + X^2 + X + 1$
31	26	1	$X^5 + X^2 + 1$
31	21	2	$X^{10} + X^9 + X^8 + X^6 + X^5 + X^3 + 1$

Reed-Solomon Codes (RS)

- Widely used subclass of nonbinary BCH codes
- Data are processed in chunks of *m*bits called symbols
- An (*n*,*k*) RS code has the following parameters:
  - Symbol length: *m* bits per symbol
  - Block length: $n = 2^m - 1$ symbols = $m(2^m - 1)$ bits
  - Data length: *k* symbols
  - Size of check code: $n - k = 2t$ symbols = $m(2t)$ bits
  - Minimum distance: $d_{min} = 2t + 1$ symbols

Parity-Check Matrix Codes

- Forward error correction (FEC)
- Low-density parity-check code (LDPC)
  - Exhibit performance in terms of bit error probability that is very close to the Shannon limit and can be efficiently implemented for high-speed use
  - Used in:
    - High-speed wireless specifications
    - 802.11n and 802.11ac Wi-Fi
    - Satellite digital television transmission
    - 10-Gbps Ethernet

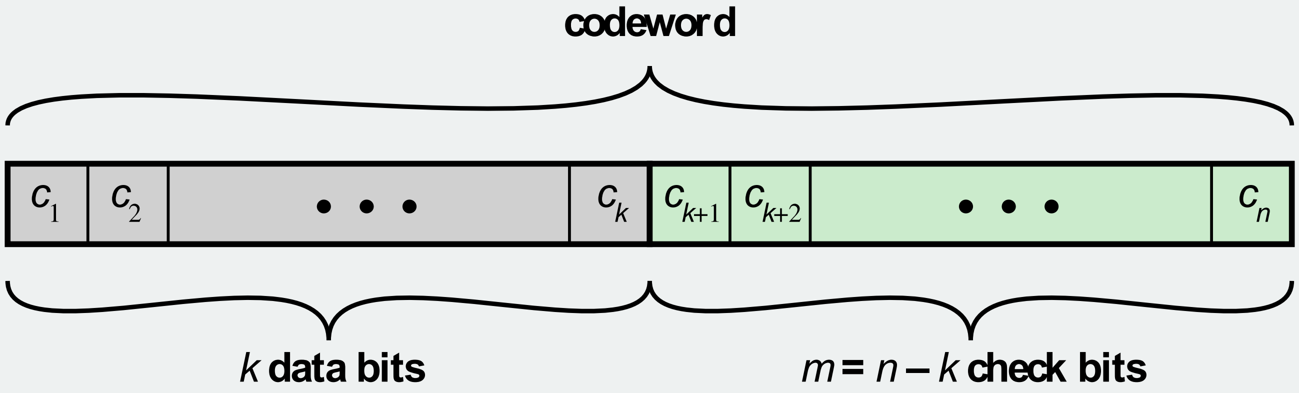


Figure 16.5 Structure of a Parity Check Codeword

Parity-Check Code

- For a parity-check code there are three functions to perform:

Encoding	For a given set of *k* data bits, generate the corresponding *n*–bit codeword
Error detection	For a given codeword, determine if there are one or more bits in error
Error correction	If an error is detected, perform error correction

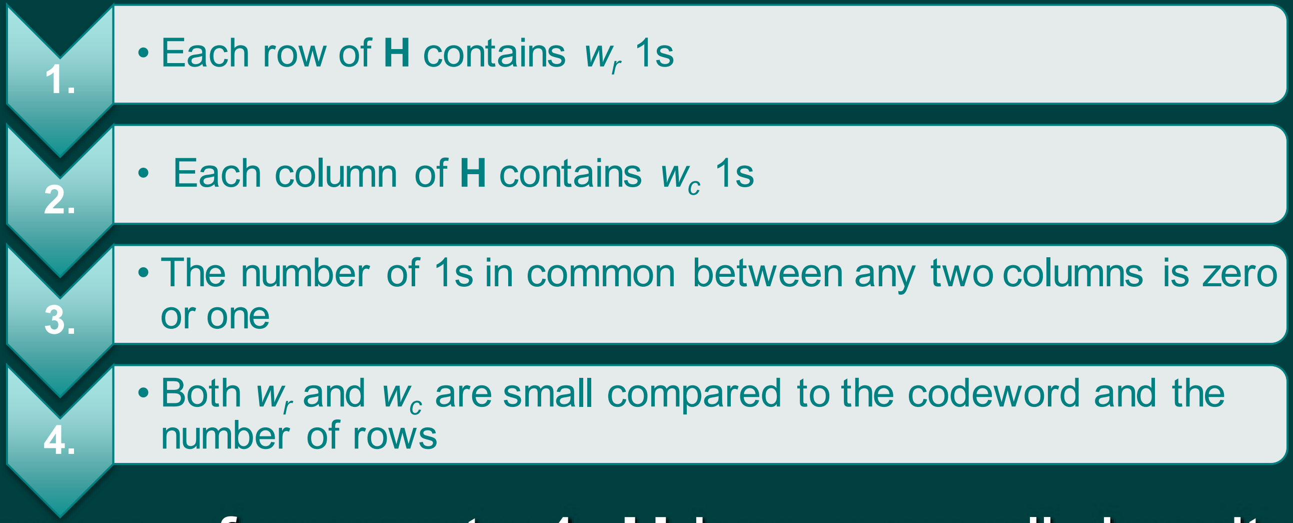
Table 16.4

(7, 4) Parity Check Code Defined by Equation (16.15)

Data bits				Check bits		
c_1	c_2	c_3	c_4	c_5	c_6	c_7
0	0	0	0	0	0	0
0	0	0	1	0	1	1
0	0	1	0	1	1	0
0	0	1	1	1	0	1
0	1	0	0	1	0	1
0	1	0	1	1	1	0
0	1	1	0	0	1	1
0	1	1	1	0	0	0
1	0	0	0	1	1	1
1	0	0	1	1	0	0
1	0	1	0	0	0	1
1	0	1	1	0	1	0
1	1	0	0	0	1	0
1	1	0	1	0	0	1
1	1	1	0	1	0	0
1	1	1	1	1	1	1

LDPC Codes

- Low-Density Parity-Check Codes
- Parity-check code with parity-check matrix **H** with the following properties:



1. Each row of **H** contains w_r 1s
2. Each column of **H** contains w_c 1s
3. The number of 1s in common between any two columns is zero or one
4. Both w_r and w_c are small compared to the codeword and the number of rows

- Because of property 4, **H** has a small density of 1s
  - The elements of **H** are almost all equal to 0
  - Hence the designation *low density*

$$\left[\begin{array}{cccccccccccc}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
\hline
1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
\hline
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1
\end{array}\right]$$

(a) Gallager parity-check matrix with $w_c = 3$, $w_r = 4$

Figure 16.6 Examples of LDPC Parity-Check Matrices

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1
\end{bmatrix}$$

(b) MacKay-Neal parity-check matrix with $w_c = 3$, $w_r = 4$

Figure 16.6 Examples of LDPC Parity-Check Matrices

$$\left[\begin{array}{ccc|ccccccccc}
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
\end{array}\right]$$

(c) Irregular repeat-accumulate parity-check matrix

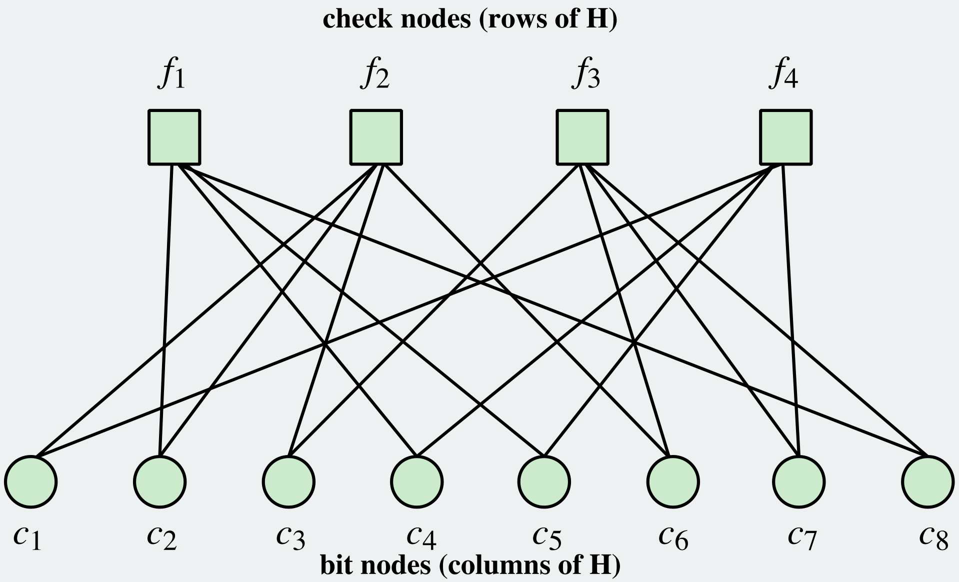
Figure 16.6 Examples of LDPC Parity-Check Matrices

Error Correction

- Error detection of an LDPC code can be performed using the parity-check matrix
  - If $\mathbf{Hc}^T$yields a nonzero vector, then an error is detected
- Tanner graph
  - Contains two kinds of nodes
    - Check nodes which correspond to rows of **H**
    - Bit nodes which correspond to columns of **H** and hence to the bits of the codeword

$$H = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

(a) Parity-check matrix



(b) Tanner Graph

Figure 16.7 Example LDPC Code

Table 16.5

Example Error Correction Technique for LDPC of Figure 16.7

(a) Constraint Equations

Row of H	Check node	Equation
1	$f1$	$c2 \oplus c4 \oplus c5 \oplus c8 = 0$
2	$f2$	$c1 \oplus c2 \oplus c3 \oplus c6 = 0$
3	$f3$	$c3 \oplus c6 \oplus c7 \oplus c8 = 0$
4	$f4$	$c1 \oplus c4 \oplus c5 \oplus c7 = 0$

(b) Messages sent and received by check nodes

Check node	Messages
$f1$	received: $c2 \rightarrow 1$ $c4 \rightarrow 1$ $c5 \rightarrow 0$ $c8 \rightarrow 1$ sent: $0 \rightarrow c2$ $0 \rightarrow c4$ $1 \rightarrow c5$ $0 \rightarrow c8$
$f2$	received: $c1 \rightarrow 1$ $c2 \rightarrow 1$ $c3 \rightarrow 0$ $c6 \rightarrow 1$ sent: $0 \rightarrow c1$ $0 \rightarrow c2$ $1 \rightarrow c3$ $0 \rightarrow c6$
$f3$	received: $c3 \rightarrow 0$ $c6 \rightarrow 1$ $c7 \rightarrow 0$ $c8 \rightarrow 1$ sent: $0 \rightarrow c3$ $1 \rightarrow c6$ $0 \rightarrow c7$ $1 \rightarrow c8$
$f4$	received: $c1 \rightarrow 1$ $c4 \rightarrow 1$ $c5 \rightarrow 0$ $c7 \rightarrow 0$ sent: $1 \rightarrow c1$ $1 \rightarrow c4$ $0 \rightarrow c5$ $0 \rightarrow c7$

(c) Estimation of codeword bit values

Bit node	Codeword bit		Messages	Decision
$c1$	1	$f2 \rightarrow 0$	$f4 \rightarrow 1$	1
$c2$	1	$f1 \rightarrow 0$	$f2 \rightarrow 0$	0
$c3$	0	$f2 \rightarrow 1$	$f3 \rightarrow 0$	0
$c4$	1	$f1 \rightarrow 0$	$f4 \rightarrow 1$	1
$c5$	0	$f1 \rightarrow 1$	$f4 \rightarrow 0$	0
$c6$	1	$f2 \rightarrow 0$	$f3 \rightarrow 1$	1
$c7$	0	$f3 \rightarrow 0$	$f4 \rightarrow 0$	0
$c8$	1	$f1 \rightarrow 0$	$f3 \rightarrow 1$	1

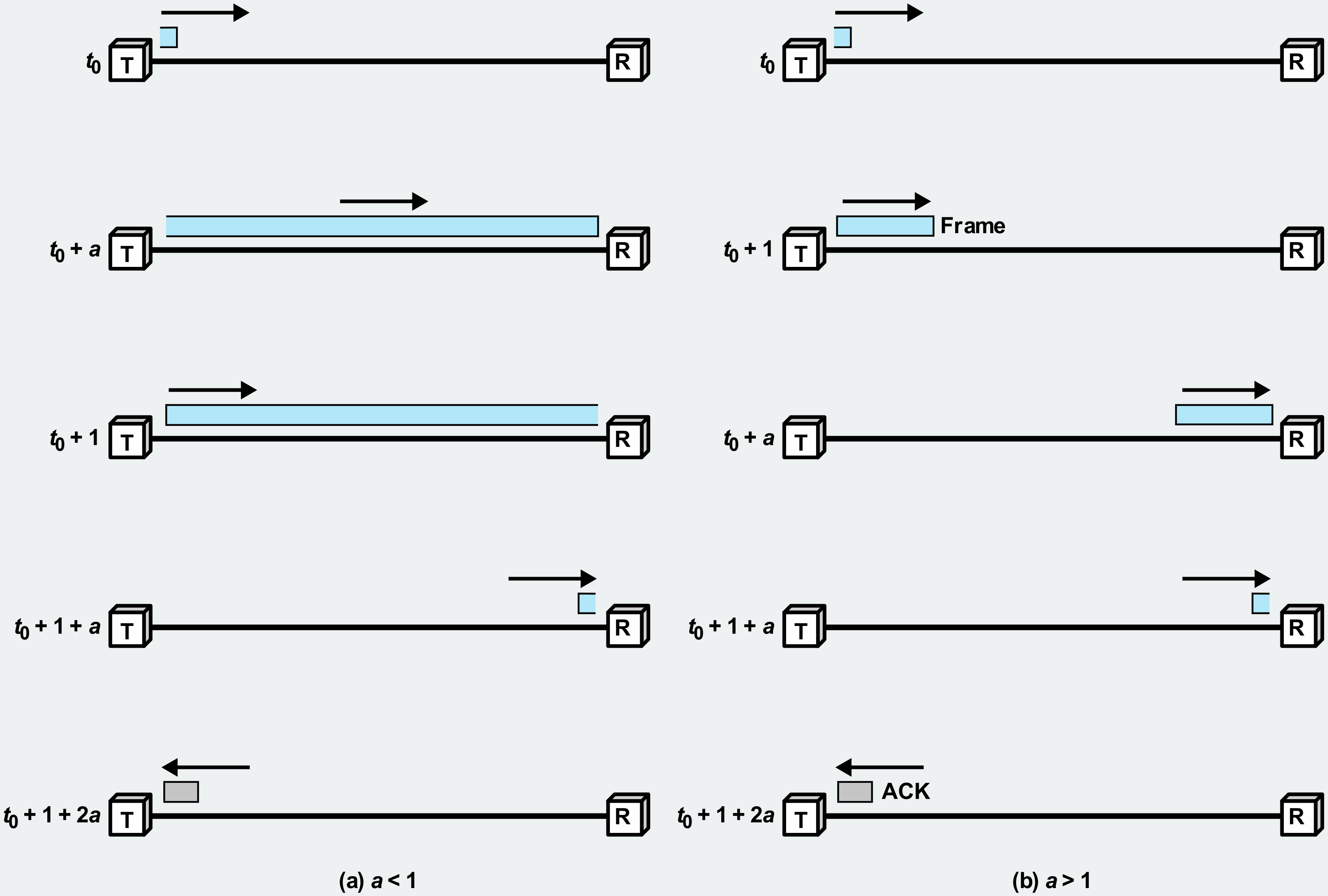


Figure 16.8 Stop-and-Wait Link Utilization (transmission time = 1; propagation time = *a*)

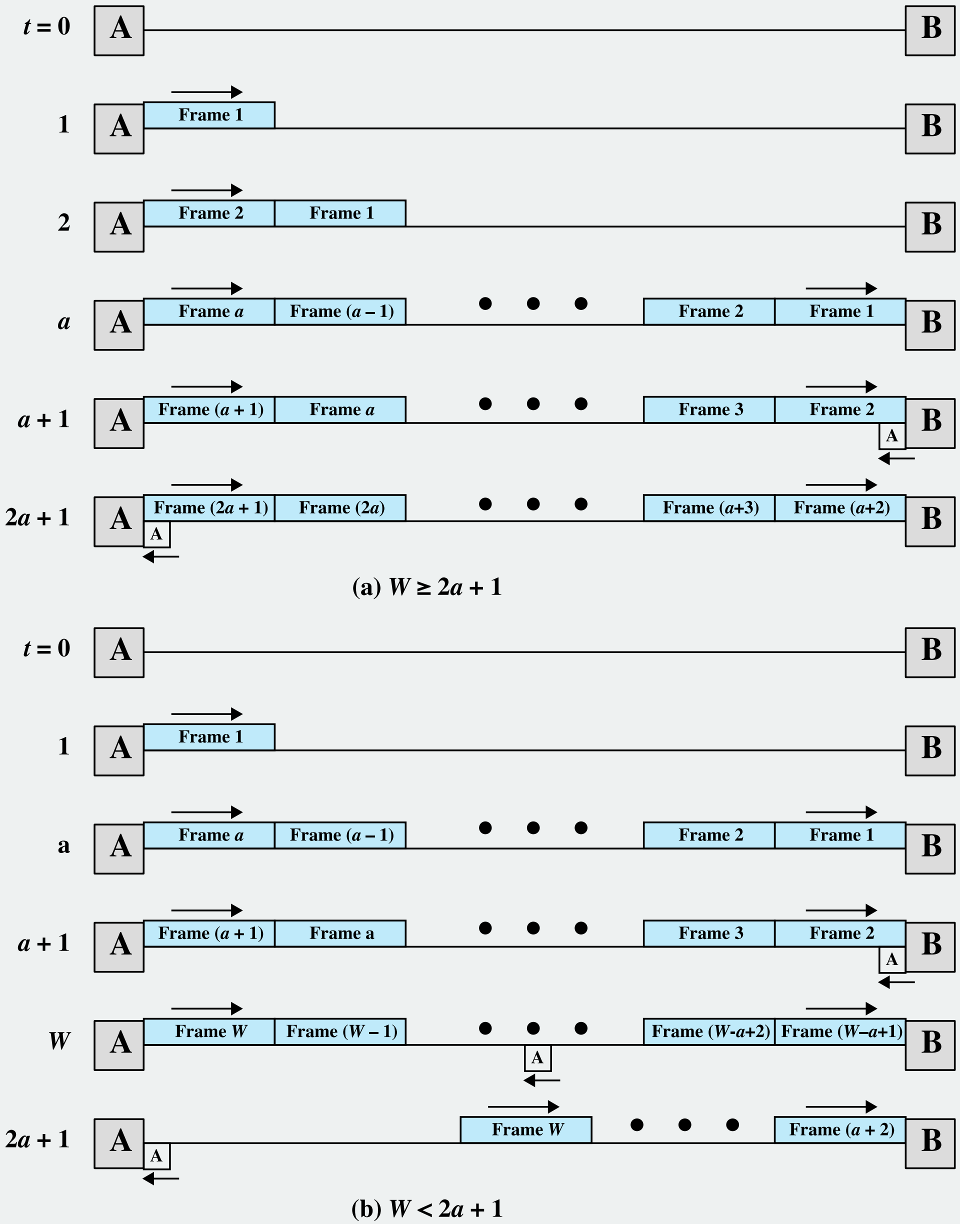


Figure 16.9 Timing of Sliding-Window Protocol

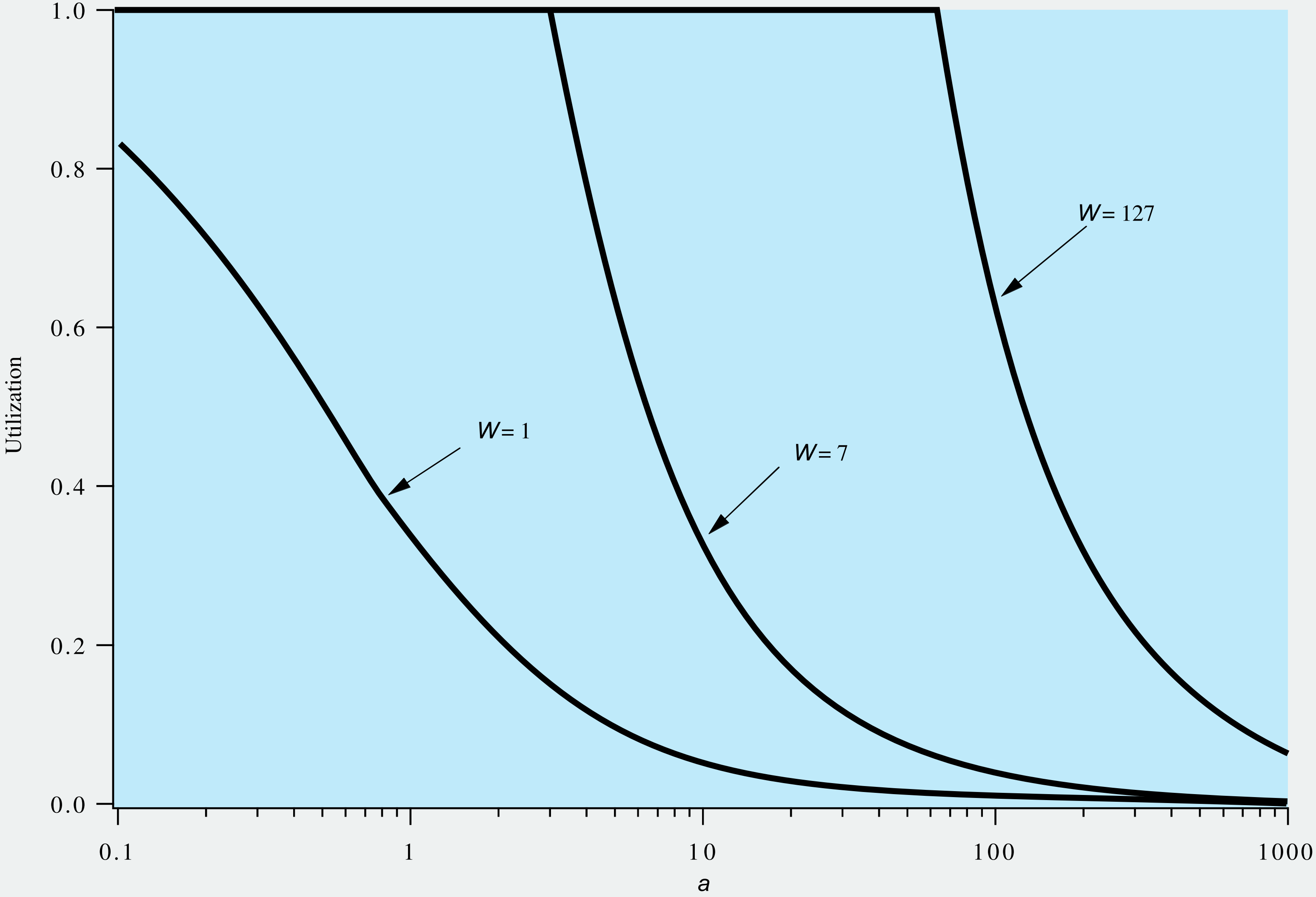


Figure 16.10 Sliding-Window Utilization as a function of *a*

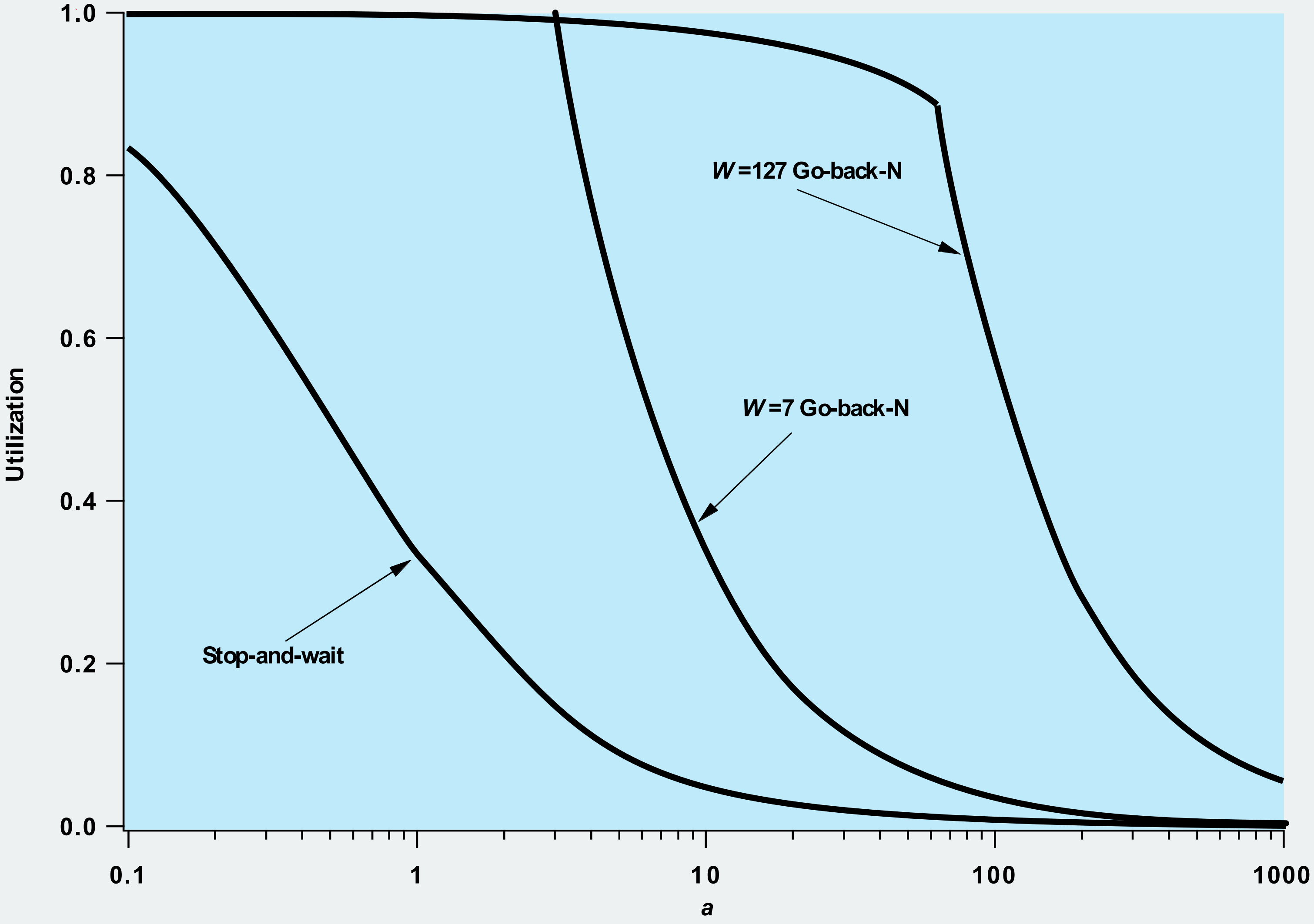


Figure 16.11 ARQ Utilization as a Function of *a* ($P = 10^{-3}$)

Summary

- Analog data, analog signals
  - Amplitude modulation
  - Angle modulation
- ARQ performance issues
  - Stop-and-wait flow control
  - Error-free sliding-window flow control
  - ARQ
- Forward error-correcting codes
  - Cyclic codes
  - BCH codes
  - Reed-Solomon codes
  - Parity-check matrix codes
  - Low-density parity-check codes