If the estimated error pattern is the same as the actual error pattern, that is, if $\hat{\mathbf{e}} = \mathbf{e}$, then the estimate $\hat{\mathbf{U}}$ is equal to the transmitted code vector $\mathbf{U}$. On the other hand, if the error estimate is incorrect, the decoder will estimate a code vector that was not transmitted, and we have an *undetectable decoding error.*

Example 5.4 Error Correction

Assume that code vector $\mathbf{U} = 1\ 0\ 1\ 1\ 1\ 0$, from the Section 5.4.3 example, is transmitted, and the vector $\mathbf{r} = 0\ 0\ 1\ 1\ 1\ 0$ is received. Show how a decoder, using the Table 5.1 syndrome look-up table, can correct the error.

Solution

The syndrome of $\mathbf{r}$ is computed:

$$\mathbf{S} = [0\ \ 0\ \ 1\ \ 1\ \ 1\ \ 0]\mathbf{H}^T = [1\ \ 0\ \ 0]$$

Using Table 5.1, the error pattern corresponding to the syndrome above is estimated to be

$$\hat{\mathbf{e}} = 1\ 0\ 0\ 0\ 0\ 0$$

The corrected vector is then estimated by

$$\begin{aligned}\hat{\mathbf{U}} &= \mathbf{r} + \hat{\mathbf{e}} \\ &= 0\ 0\ 1\ 1\ 1\ 0 + 1\ 0\ 0\ 0\ 0\ 0 \\ &= 1\ 0\ 1\ 1\ 1\ 0\end{aligned}$$

Since the estimated error pattern is the actual error pattern in this example, the error correction procedure yields $\hat{\mathbf{U}} = \mathbf{U}$.

5.5 CODING STRENGTH

5.5.1 Weight and Distance of Binary Vectors

It should be clear that not all error patterns can be correctly decoded. The error correction capability of a code will be investigated by first defining its structure. The *Hamming weight*, $w(\mathbf{U})$, of a vector $\mathbf{U}$ is defined to be the number of nonzero elements in $\mathbf{U}$. For a binary vector this is equivalent to the number of ones in the vector. For example, if $\mathbf{U} = 1\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1$, then $w(\mathbf{U}) = 5$. The *Hamming distance* between two code vectors $\mathbf{U}$ and $\mathbf{V}$, denoted $d(\mathbf{U}, \mathbf{V})$, is defined to be the number of elements in which they differ: for example,

$$\begin{aligned}\mathbf{U} &= 1\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1 \\ \mathbf{V} &= 0\ 1\ 1\ 1\ 1\ 0\ 1\ 0\ 0 \\ d(\mathbf{U}, \mathbf{V}) &= 6\end{aligned}$$

By the properties of modulo-2 addition, we note that the sum of two binary vectors

is another vector whose binary ones are located in those positions in which the two vectors differ: for example,

$$\mathbf{U} + \mathbf{V} = 1\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 1$$

Thus we observe that the Hamming distance between two code vectors is equal to the Hamming weight of their sum: that is, $d(\mathbf{U}, \mathbf{V}) = w(\mathbf{U} + \mathbf{V})$. Also, we see that the Hamming weight of a code vector is equal to its Hamming distance from the all-zeros vector.

5.5.2 Minimum Distance of a Linear Code

Consider the set of distances between all pairs of code vectors in the space V_n. The smallest member of the set is the *minimum distance* of the code and is denoted $d_{\min}$. Why do you suppose we have an interest in the minimum distance; why not the maximum distance? The minimum distance, like the weakest link in a chain, gives us a measure of the code's minimum capability and therefore characterizes the code's strength.

As discussed earlier, the sum of any two code vectors yields another code vector member of the subspace. This property of linear codes is stated simply as: If $\mathbf{U}$ and $\mathbf{V}$ are code vectors, then $\mathbf{W} = \mathbf{U} + \mathbf{V}$ must also be a code vector. Hence the distance between two code vectors is equal to the weight of a third code vector; that is, $d(\mathbf{U}, \mathbf{V}) = w(\mathbf{U} + \mathbf{V}) = w(\mathbf{W})$. Thus the minimum distance of a linear code can be ascertained without examining the distance between all combinations of code vector pairs. We only need to examine the weight of each code vector (excluding the all-zeros vector) in the subspace; the minimum weight corresponds to the minimum distance, $d_{\min}$. Equivalently, $d_{\min}$ corresponds to the smallest of the set of distances between the all-zeros code vector and all the other code vectors.

5.5.3 Error Detection and Correction

The task of the decoder, having received the vector $\mathbf{r}$, is to estimate the transmitted code vector $\mathbf{U}_i$. The optimal decoder strategy can be expressed in terms of the *maximum likelihood* algorithm (see Appendix B) as follows: Decide in favor of $\mathbf{U}_i$ if

$$P(\mathbf{r}|\mathbf{U}_i) = \max_{\text{over all } \mathbf{U}_j} P(\mathbf{r}|\mathbf{U}_j) \tag{5.41}$$

Since for the binary symmetric channel (BSC), the likelihood of $\mathbf{U}_i$ with respect to $\mathbf{r}$ is inversely proportional to the distance between $\mathbf{r}$ and $\mathbf{U}_i$, we can write: Decide in favor of $\mathbf{U}_i$ if

$$d(\mathbf{r}, \mathbf{U}_i) = \min_{\text{over all } \mathbf{U}_j} d(\mathbf{r}, \mathbf{U}_j) \tag{5.42}$$

In other words, the decoder determines the distance between $\mathbf{r}$ and each of the possible transmitted code vectors $\mathbf{U}_i$, and selects as most likely a $\mathbf{U}_i$ for which

$$d(\mathbf{r}, \mathbf{U}_i) \leq d(\mathbf{r}, \mathbf{U}_j) \qquad \text{for } i, j = 1, \ldots, M \quad \text{and} \quad i \neq j \tag{5.43}$$

where $M = 2^k$ is the size of the code vector set. If the minimum is not unique, the choice between minimum distance codewords is arbitrary. Distance metrics are treated further in Chapter 6.

In Figure 5.15 the distance between two code vectors $\mathbf{U}$ and $\mathbf{V}$ is shown using a number line calibrated in *Hamming distance*. Each black dot represents a corrupted code vector. Figure 5.15a illustrates the reception of vector $\mathbf{r}_1$, which is distance 1 from $\mathbf{U}$ and distance 4 from $\mathbf{V}$. An error-correcting decoder, following the maximum likelihood strategy, will select $\mathbf{U}$ upon receiving $\mathbf{r}_1$. If $\mathbf{r}_1$ had been the result of a 1-bit corruption to the transmitted code vector $\mathbf{U}$, the decoder has successfully corrected the error. But if $\mathbf{r}_1$ had been the result of a 4-bit corruption to the transmitted code vector $\mathbf{V}$, the result is a decoding error. Similarly, a double error in transmission of $\mathbf{U}$ might result in the received vector $\mathbf{r}_2$, which is distance 2 from $\mathbf{U}$ and distance 3 from $\mathbf{V}$, as shown in Figure 5.15b. Here, too, the decoder will select $\mathbf{U}$ upon receiving $\mathbf{r}_2$. A triple error in transmission of $\mathbf{U}$ might result in a received vector $\mathbf{r}_3$ which is distance 3 from $\mathbf{U}$ and distance 2 from $\mathbf{V}$, as shown in Figure 5.15c. Here the decoder will select $\mathbf{V}$ upon receiving $\mathbf{r}_3$, and will have made an error in decoding. From Figure 5.15 it should be clear that if error

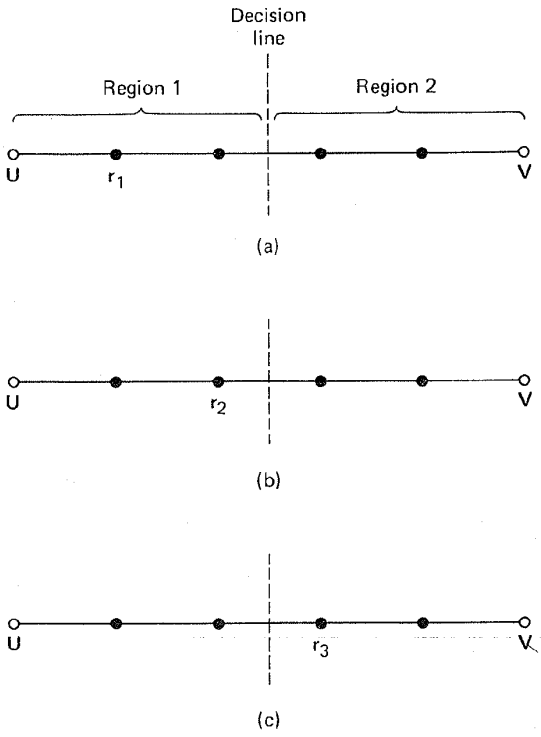


Figure 5.15 Error correction and detection strength. (a) Received vector $\mathbf{r}_1$. (b) Received vector $\mathbf{r}_2$. (c) Received vector $\mathbf{r}_3$.

detection and not correction is the task, a corrupted vector, characterized by a black dot and representing a 1-bit, 2-bit, 3-bit, or 4-bit error, can be detected. However, five errors in transmission might result in code vector **V** being received when code vector **U** was actually transmitted; such an error would be *undetectable*.

From Figure 5.15 we can see that the error-detecting and error-correcting capabilities of a code are related to the *minimum distance* between code vectors. The decision line in the figure serves the same purpose in the process of decoding as it does in demodulation, to define the decision regions. In the Figure 5.15 example, the decision criterion of choosing **U** if **r** falls in region 1, and choosing **V** if **r** falls in region 2, illustrates that such a code, with $d_{\min} = 5$, can correct two errors. In general, the *error-correcting capability*, t, of a code is defined as the maximum number of guaranteed correctable errors per codeword, and is written [4]

$$t = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor \tag{5.44}$$

where $\lfloor x \rfloor$ means the largest integer not to exceed x. Often, a code that corrects all possible sequences of t or fewer errors can also correct certain sequences of $t + 1$ errors. This can be seen in Figure 5.14. In this example $d_{\min} = 3$, and thus from Equation (5.44), we can see that *all* $t = 1$ bit-error patterns are correctable. Also, *a single* $t + 1$ or 2-bit error pattern is correctable. In general, a t-error-correcting (n, k) linear code is capable of correcting a total of 2^{n-k} error patterns. If a t-error-correcting block code is used strictly for error correction on a binary symmetric channel (BSC) with transition probability p, the probability that the decoder commits an erroneous decoding, and that the n-bit block is in error, can be calculated by using Equation (5.18) as an upper bound:

$$P_M \leq \sum_{j=t+1}^{n} \binom{n}{j} p^j (1 - p)^{n-j} \tag{5.45}$$

The bound becomes an equality when the decoder corrects all combinations of errors up to and including t errors, but no combinations of errors greater than t. Such decoders are called *bounded distance decoders*. The decoded bit-error probability depends on the particular code and decoder. It can be expressed [5] by the following approximation:

$$P_B \simeq \frac{1}{n} \sum_{j=t+1}^{n} j \binom{n}{j} p^j (1 - p)^{n-j} \tag{5.46}$$

A code can be used to detect errors prior to, or instead of, correcting them. It should be clear from Figure 5.15 that any received vector characterized by a black dot (a corrupted code vector) can be identified as an error. Therefore, the error-detecting capability, e, is defined in terms of $d_{\min}$ as

$$e = d_{\min} - 1 \tag{5.47}$$

A block code with minimum distance $d_{\min}$ guarantees that all error patterns of

$d_{\min} - 1$ or fewer errors can be detected. Such a code is also capable of detecting a large fraction of error patterns with $d_{\min}$ or more errors. In fact, an (n, k) code is capable of detecting $2^n - 2^k$ error patterns of length n. The reasoning is as follows. There are a total of $2^n - 1$ possible nonzero error patterns in the space of 2^n n-tuples. Even the bit pattern of a valid codeword represents a potential error pattern. Thus there are $2^k - 1$ error patterns that are identical to the $2^k - 1$ nonzero codewords. If any of these $2^k - 1$ error patterns occurs, it alters the transmitted codeword $\mathbf{U}_i$ into another codeword $\mathbf{U}_j$. Thus $\mathbf{U}_j$ will be received and its syndrome is zero. The decoder accepts $\mathbf{U}_j$ as the transmitted codeword and thereby commits an incorrect decoding. Therefore, there are $2^k - 1$ undetectable error patterns. If the error pattern is not identical to one of the 2^k codewords, the syndrome test on the received vector $\mathbf{r}$ yields a nonzero syndrome, and the error is detected. Therefore, there are exactly $2^n - 2^k$ detectable error patterns. For large n, where $2^k \ll 2^n$, only a small fraction of error patterns are undetected.

5.5.3.1 Code Vector Weight Distribution

Let A_j be the number of code vectors of weight j within an (n, k) linear code. The numbers $A_0, A_1, \ldots, A_n$ are called the *weight distribution* of the code. If the code is used only for error detection, on a BSC, the probability, P_{nd}, that the decoder does not detect an error can be computed from the weight distribution of the code [5] as follows:

$$P_{\text{nd}} = \sum_{j=1}^{n} A_j p^j (1 - p)^{n-j} \tag{5.48}$$

where p is the transition probability of the BSC. If the minimum distance of the code is $d_{\min}$, the values of A_1 to $A_{d_{\min}-1}$ are zero.

Example 5.5 Probability of an Undetected Error in an Error Detecting Code

Consider that the (6, 3) code, given in Section 5.4.3, is used only for error detection. Calculate the probability of an undetected error if the channel is a BSC and the transition probability is 10^{-2}.

Solution

The weight distribution of this code is $A_0 = 1$, $A_1 = A_2 = 0$, $A_3 = 4$, $A_4 = 3$, $A_5 = 0$, $A_6 = 0$. Therefore, we can write, using Equation (5.48),

$$P_{\text{nd}} = 4p^3(1 - p)^3 + 3p^4(1 - p)^2$$

For $p = 10^{-2}$, the probability of an undetected error is 3.9×10^{-6}.

5.5.3.2 Simultaneous Error Correction and Detection

It is possible to trade correction capability from the maximum guaranteed (t), where t is defined in Equation (5.44), for the ability to simultaneously detect a class of errors. A code can be used for the simultaneous correction of α errors and detection of β errors where $\beta \geq \alpha$, provided that its minimum distance is [4]

$$d_{\min} \geq \alpha + \beta + 1 \tag{5.49}$$

When t or fewer errors occur, the code is capable of detecting and correcting them. When more than t but fewer than $e + 1$ errors occur, where e is defined in Equation (5.47), the code is capable of detecting their presence but correcting only a subset of them. For example, a code with $d_{\min} = 7$ can be used to simultaneously detect and correct in any one of the following ways:

Detect (β)	Correct (α)
3	3
4	2
5	1
6	0

Note that correction implies prior detection. For the above example, when there are three errors, all of them can be detected and corrected. When there are five errors, all of them can be detected but only a subset of them (one) can be corrected.

5.5.4 Visualization of a 6-Tuple Space

Figure 5.16 is a visualization of the eight codewords from the example of Section 5.4.3. The codewords are generated from linear combinations of the three independent 6-tuples in Equation (5.26); the codewords form a three-dimensional subspace. The figure shows such a subspace completely occupied by the eight codewords (large black circles); the coordinates of the subspace have purposely been drawn to emphasize their nonorthogonality. Figure 5.16 is an attempt to illustrate the entire space, containing sixty-four 6-tuples, even though there is no precise way to draw or construct such a model. Spherical layers or shells are shown around each codeword. Each of the nonintersecting inner layers is a Hamming distance of 1 from its associated codeword; each outer layer is a Hamming distance of 2 from its codeword. Larger distances are not useful in this example. For each codeword, the two layers shown are occupied by perturbed codewords. There are six such points on each inner sphere (a total of 48 points), representing the six possible 1-bit error-perturbed vectors associated with each codeword. These 1-bit perturbed codewords are distinct in the sense that they can best be associated with only one codeword, and therefore can be corrected. As is seen from the standard array of Figure 5.14, there is also one 2-bit error pattern that can be corrected. There is a total of $\binom{6}{2} = 15$ different 2-bit error patterns that can be inflicted on each codeword, but only one of them, in our example the 0 1 0 0 0 1 error pattern, can be corrected. The other fourteen 2-bit error patterns yield vectors that cannot be uniquely identified with just one codeword; these noncorrectable error patterns yield vectors that are equivalent to the error-perturbed vectors of two or more codewords. In the figure, all correctable (fifty-six) 1- and 2-bit error-perturbed codewords are shown as small black circles. Perturbed codewords that cannot be corrected are shown as small clear circles.

Figure 5.16 is useful for visualizing the properties of a class of codes known as *perfect codes*. A t-error-correcting code is called a perfect code if its standard array has all the error patterns of t or fewer errors and no others as coset leaders.

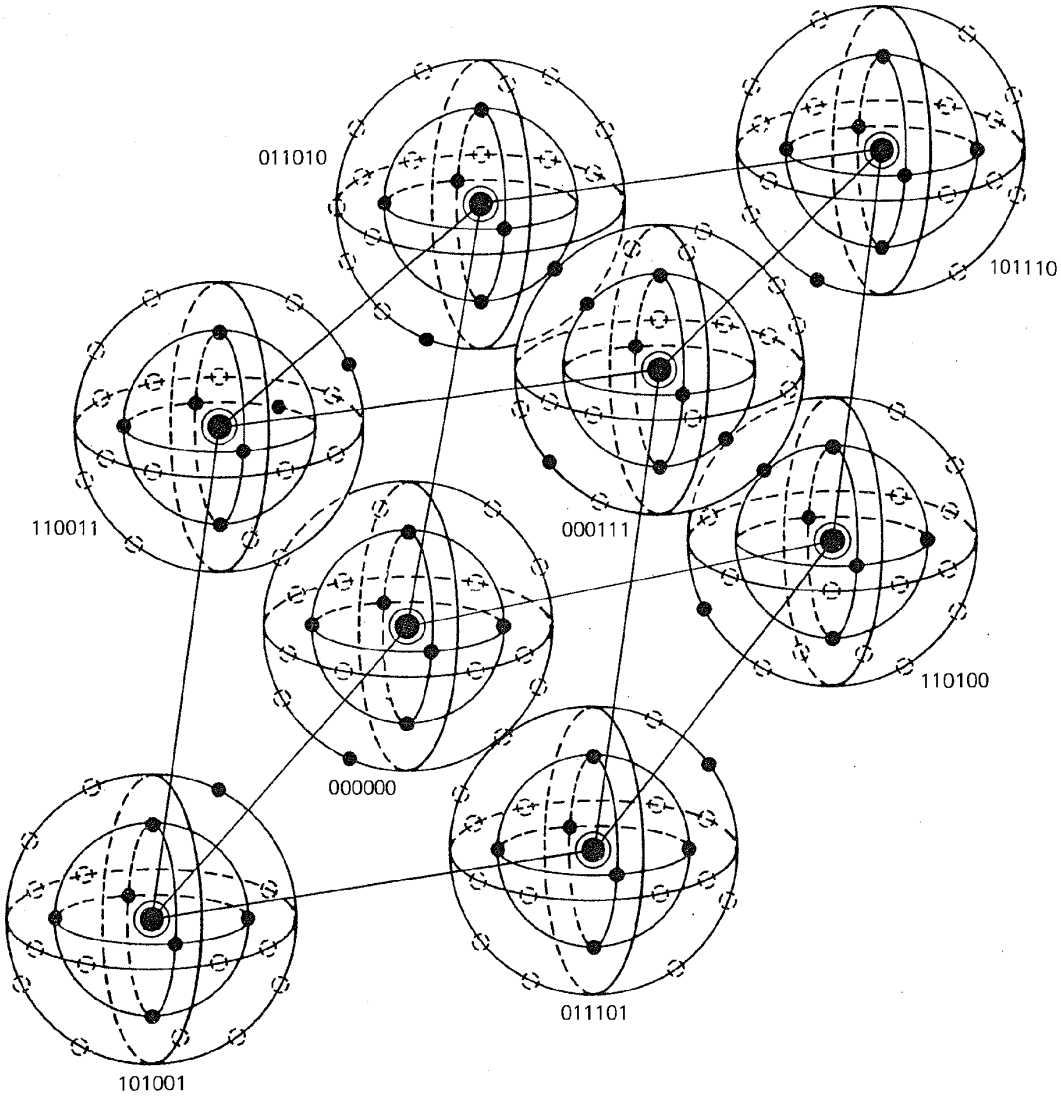


Figure 5.16 Example of eight codewords in a 6-tuple space.

In terms of Figure 5.16, a t-error-correcting perfect code is one that can, with maximum likelihood decoding, correct all perturbed code vectors occupying a shell at Hamming distance t or less from its originating codeword, and cannot correct any perturbed vectors occupying shells at distances greater than t.

Figure 5.16 is also useful for understanding the basic goal in the search for good codes. We would like for the space to be filled with as many codewords as possible (efficient utilization of the added redundancy), and we would also like

these codewords to be as far away from one another as possible. Obviously, these goals conflict.

5.5.5 Erasure Correction

A receiver may be designed to declare a symbol *erased* when it is received ambiguously or when the receiver recognizes the presence of interference or a transient malfunction. Such a channel has an input alphabet of size Q and an output alphabet of size $Q + 1$; the extra output symbol is called an *erasure flag*, or simply an *erasure*. When a demodulator makes a symbol error, two parameters are needed to correct that error, its *location* and its *correct* symbol value. In the case of binary symbols, this reduces to needing only the error location. However, if the demodulator declares a symbol *erased*, although the correct symbol value is not known, the symbol location *is* known, and for this reason, the decoding of erased codewords can be simpler than error correcting. An error control code can be used to correct erasures or to correct errors and erasures simultaneously. If the code has minimum distance $d_{\min}$, any pattern of ρ or fewer erasures can be corrected if [6]

$$d_{\min} \geq \rho + 1 \tag{5.50}$$

Assume for the moment that no errors occur outside the erasure positions. The advantage of correcting by means of erasures is expressed quantitatively as follows: If a code has a minimum distance $d_{\min}$, then from Equation (5.50), $d_{\min} - 1$ erasures can be reconstituted. Since the number of errors that can be corrected without erasure information is $(d_{\min} - 1)/2$ at most, from Equation (5.44), the advantage of correcting by means of erasures is clear. Further, any pattern of α errors and γ erasures can be corrected simultaneously if [6]

$$d_{\min} \geq 2\alpha + \gamma + 1 \tag{5.51}$$

Simultaneous erasure correction and error correction can be accomplished in the following way. First, the γ-erased positions are replaced with zeros and the resulting codeword is decoded normally. Next, the γ-erased positions are replaced with ones, and the decoding operation is repeated on this version of the codeword. Of the two codewords obtained (one with erasures replaced by zeros, and the other with erasures replaced by ones) the one corresponding to the smallest number of errors corrected outside the γ-erased positions is selected. This technique will always result in correct decoding if Equation (5.51) is satisfied.

Example 5.6 Erasure Correction

Consider the codeword set presented in Section 5.4.3:

000000 110100 011010 101110 101001 011101 110011 000111

Suppose that the codeword 110011 was transmitted and that the two leftmost digits were declared by the receiver to be erasures. Verify that the received flawed sequence xx0011 can be corrected.

Solution

Since $d_{\min} = \rho + 1 = 3$, the code can correct as many as $\rho = 2$ erasures. This is easily verified above or with Figure 5.14 by comparing the rightmost four digits of xx0011 with each of the allowable codewords. The codeword that was actually transmitted is closest in Hamming distance to the flawed sequence.

5.6 CYCLIC CODES

Binary cyclic codes are an important subclass of linear block codes. The codes are easily implemented with feedback shift registers; the syndrome calculation is easily accomplished with similar feedback shift registers; and the underlying algebraic structure of a cyclic code lends itself to efficient decoding methods. An (n, k) linear code is called a *cyclic code* if it can be described by the following property. If the n-tuple $\mathbf{U} = (u_0, u_1, u_2, \ldots, u_{n-1})$ is a code vector in the subspace S, then $\mathbf{U}^{(1)} = (u_{n-1}, u_0, u_1, \ldots, u_{n-2})$ obtained by an end-around shift, is also a code vector in S. Or in general, $\mathbf{U}^{(i)} = (u_{n-i}, u_{n-i+1}, \ldots, u_{n-1}, u_0, u_1, \ldots, u_{n-i-1})$, obtained by i end-around or cyclic shifts, is also a code vector in S.

The components of a code vector $\mathbf{U} = (u_0, u_1, u_2, \ldots, u_{n-1})$ can be treated as the coefficients of a polynomial $\mathbf{U}(X)$ as follows:

$$\mathbf{U}(X) = u_0 + u_1X + u_2X^2 + \cdots + u_{n-1}X^{n-1} \tag{5.52}$$

The polynomial function $\mathbf{U}(X)$ can be thought of as a "placeholder" for the digits of the code vector $\mathbf{U}$; that is, an n-tuple vector is described by a polynomial of degree $n - 1$ or less. The presence or absence of each term in the polynomial indicates the presence of a 1 or 0 in the corresponding location of the n-tuple. If the u_{n-1} component is nonzero, the polynomial is of degree $n - 1$. The usefulness of this polynomial description of a codeword will become clear as we discuss the algebraic structure of the cyclic codes.

5.6.1 Algebraic Structure of Cyclic Codes

Expressing the code vectors in polynomial form, the cyclic nature of the code manifests itself in the following way. If $\mathbf{U}(X)$ is an $(n - 1)$-degree codeword polynomial, then $\mathbf{U}^{(i)}(X)$, the remainder resulting from dividing $X^i\mathbf{U}(X)$ by $X^n + 1$, is also a codeword; that is,

$$\frac{X^i\mathbf{U}(X)}{X^n + 1} = \mathbf{q}(X) + \frac{\mathbf{U}^{(i)}(X)}{X^n + 1} \tag{5.53}$$

or, multiplying through by $X^n + 1$,

$$X^i\mathbf{U}(X) = \mathbf{q}(X)(X^n + 1) \underbrace{+ \mathbf{U}^{(i)}(X)}_{\text{remainder}} \tag{5.54}$$

which can also be described in terms of modulo arithmetic as follows:

$$\mathbf{U}^{(i)}(X) = X^i\mathbf{U}(X) \text{ modulo } (X^n + 1) \tag{5.55}$$

where x modulo y is defined as the remainder obtained from dividing x by y. Let us demonstrate the validity of Equation (5.55) for the case of $i = 1$.

$$\mathbf{U}(X) = u_0 + u_1X + u_2X^2 + \cdots + u_{n-2}X^{n-2} + u_{n-1}X^{n-1}$$

$$X\mathbf{U}(X) = u_0X + u_1X^2 + u_2X^3 + \cdots + u_{n-2}X^{n-1} + u_{n-1}X^n$$

We now add and subtract u_{n-1}, or since we are using modulo-2 arithmetic, we add u_{n-1} twice, as follows:

$$\underbrace{X\mathbf{U}(X) = u_{n-1} + u_0X + u_1X^2 + u_2X^3 + \cdots + u_{n-2}X^{n-1}}_{\mathbf{U}^{(1)}(X)} + u_{n-1}X^n + u_{n-1}$$

$$= \mathbf{U}^{(1)}(X) + u_{n-1}(X^n + 1)$$

Since $\mathbf{U}^{(1)}(X)$ is of degree $n - 1$, it cannot be divided by $X^n + 1$. Thus we can write from Equation (5.53)

$$\mathbf{U}^{(1)}(X) = X\mathbf{U}(X) \text{ modulo } (X^n + 1)$$

By extension we can write

$$\mathbf{U}^{(i)}(X) = X^i\mathbf{U}(X) \text{ modulo } (X^n + 1) \tag{5.56}$$

Example 5.7 Cyclic Shift of a Code Vector

Let $\mathbf{U} = 1\ 1\ 0\ 1$, for $n = 4$. Express the code vector in polynomial form, and using Equation (5.54), solve for the third end-around shift of the code vector.

Solution

$\mathbf{U}(X) = 1 + X + X^3$ (polynomial is written low order to high order)

$X^i\mathbf{U}(X) = X^3 + X^4 + X^6$ where $i = 3$

Divide $X^3\mathbf{U}(X)$ by $X^4 + 1$, and solve for the remainder using polynomial division.

$$\begin{array}{r} X^2 + 1 \\ X^4 + 1 \;\overline{)\,X^6 + X^4 + X^3} \\ \underline{X^6 \qquad\quad + X^2} \\ X^4 + X^3 + X^2 \\ \underline{X^4 \qquad\quad + 1} \\ X^3 + X^2 + 1 \end{array} \qquad \text{remainder } \mathbf{U}^{(3)}(X)$$

Writing the remainder low order to high order: $1 + X^2 + X^3$, the codeword $\mathbf{U}^{(3)} = 1\ 0\ 1\ 1$ is three cyclic shifts of $\mathbf{U} = 1\ 1\ 0\ 1$. Remember that for binary codes, the addition operation is performed modulo-2, so that $+1 = -1$, and we consequently do not show any minus signs in the computation.

5.6.2 Binary Cyclic Code Properties

We can generate a cyclic code using a *generator polynomial* in much the way that we generated a block code using a generator matrix. The generator polynomial $\mathbf{g}(X)$ for an (n, k) cyclic code is unique and is of the form

$$\mathbf{g}(X) = g_0 + g_1X + g_2X^2 + \cdots + g_rX^r \tag{5.57}$$

where g_0 and g_r must equal 1. Every codeword polynomial in the subspace is of the form $\mathbf{U}(X) = \mathbf{m}(X)\mathbf{g}(X)$, where $\mathbf{U}(X)$ is a polynomial of degree $n - 1$ or less. Therefore, the message polynomial $\mathbf{m}(X)$ is written

$$\mathbf{m}(X) = m_0 + m_1X + m_2X^2 + \cdots + m_{n-r-1}X^{n-r-1} \tag{5.58}$$

There are 2^{n-r} codeword polynomials, and there are 2^k code vectors in an (n, k) code. Since there must be one codeword polynomial for each code vector

$$n - r = k$$

or

$$r = n - k$$

Hence $\mathbf{g}(X)$, as shown in Equation (5.57), must be of degree $n - k$, and every codeword polynomial in the (n, k) cyclic code can be expressed as

$$\mathbf{U}(X) = (m_0 + m_1X + m_2X^2 + \cdots + m_{k-1}X^{k-1})\mathbf{g}(X) \tag{5.59}$$

U is said to be a valid code vector of the subspace S *if, and only if,* $\mathbf{g}(X)$ divides into $\mathbf{U}(X)$ without a remainder.

A generator polynomial $\mathbf{g}(X)$ of an (n, k) cyclic code is a factor of $X^n + 1$; that is, $X^n + 1 = \mathbf{g}(X)\mathbf{h}(X)$. For example,

$$X^7 + 1 = (1 + X + X^3)(1 + X + X^2 + X^4)$$

Using $\mathbf{g}(X) = 1 + X + X^3$ as a generator polynomial of degree $n - k = 3$, we can generate an $(n, k) = (7, 4)$ cyclic code. Or, using $\mathbf{g}(X) = 1 + X + X^2 + X^4$ where $n - k = 4$ we can generate a (7, 3) cyclic code. In summary, if $\mathbf{g}(X)$ is a polynomial of degree $n - k$ and is a factor of $X^n + 1$, then $\mathbf{g}(X)$ uniquely generates an (n, k) cyclic code.

5.6.3 Encoding in Systematic Form

In Section 5.4.5 we introduced the *systematic* form and discussed the reduction in complexity that makes this encoding form attractive. Let us use some of the algebraic properties of the cyclic code to establish a systematic encoding procedure. We can express the message vector in polynomial form, as follows:

$$\mathbf{m}(X) = m_0 + m_1X + m_2X^2 + \cdots + m_{k-1}X^{k-1} \tag{5.60}$$

In systematic form, the message digits are utilized as part of the code vector. We

can think of shifting the message digits into the rightmost k stages of a codeword register, and then appending the parity digits by placing them in the leftmost $n - k$ stages. Therefore, we want to manipulate the message polynomial algebraically so that it is right-shifted $n - k$ positions. If we multiply $\mathbf{m}(X)$ by X^{n-k} we get the right-shifted message polynomial:

$$X^{n-k}\mathbf{m}(X) = m_0X^{n-k} + m_1X^{n-k+1} + \cdots + m_{k-1}X^{n-1} \tag{5.61}$$

If we next divide Equation (5.61) by $\mathbf{g}(X)$, the result can be expressed as

$$X^{n-k}\mathbf{m}(X) = \mathbf{q}(X)\mathbf{g}(X) + \mathbf{r}(X) \tag{5.62}$$

where

$$\mathbf{r}(X) = r_0 + r_1X + r_2X^2 + \cdots + r_{n-k-1}X^{n-k-1}$$

We can also say that

$$\mathbf{r}(X) = X^{n-k}\mathbf{m}(X) \text{ modulo } \mathbf{g}(X) \tag{5.63}$$

Adding $\mathbf{r}(X)$ to both sides of Equation (5.62), using modulo-2 arithmetic, we get

$$\mathbf{r}(X) + X^{n-k}\mathbf{m}(X) = \mathbf{q}(X)\mathbf{g}(X) = \mathbf{U}(X) \tag{5.64}$$

The left-hand side of Equation (5.64) is recognized as a valid codeword polynomial, since it is a polynomial of degree $n - 1$ or less, and when divided by $\mathbf{g}(X)$ there is a zero remainder. This codeword can be expanded into its polynomial terms as follows:

$$\mathbf{r}(X) + X^{n-k}\mathbf{m}(X) = r_0 + r_1X + \cdots + r_{n-k-1}X^{n-k-1} + m_0X^{n-k} + m_1X^{n-k+1} + \cdots + m_{k-1}X^{n-1}$$

The codeword polynomial corresponds to the code vector

$$\mathbf{U} = \underbrace{(r_0, r_1, \ldots, r_{n-k-1})}_{(n-k) \text{ parity bits}}, \underbrace{m_0, m_1, \ldots, m_{k-1})}_{k \text{ message bits}} \tag{5.65}$$

Example 5.8 Cyclic Code in Systematic Form

Using the generator polynomial $\mathbf{g}(X) = 1 + X + X^3$, generate a systematic code vector from the (7, 4) codeword set for the message vector $\mathbf{m} = 1\ 0\ 1\ 1$.

Solution

$$\mathbf{m}(X) = 1 + X^2 + X^3, \quad n = 7, \quad k = 4, \quad n - k = 3$$

$$X^{n-k}\mathbf{m}(X) = X^3(1 + X^2 + X^3) = X^3 + X^5 + X^6$$

Dividing $X^{n-k}\mathbf{m}(X)$ by $\mathbf{g}(X)$ using polynomial division, we can write

$$X^3 + X^5 + X^6 = \underbrace{(1 + X + X^2 + X^3)}_{\substack{\text{quotient} \\ \mathbf{q}(X)}} \underbrace{(1 + X + X^3)}_{\substack{\text{generator} \\ \mathbf{g}(X)}} + \underbrace{1}_{\substack{\text{remainder} \\ \mathbf{r}(X)}}$$

Using Equation (5.64) yields

$$\mathbf{U}(X) = \mathbf{r}(X) + X^3\mathbf{m}(X) = 1 + X^3 + X^5 + X^6$$

$$\mathbf{U} = \underbrace{1\ 0\ 0}_{\substack{\text{parity}\\ \text{bits}}} \quad \underbrace{1\ 0\ 1\ 1}_{\substack{\text{message}\\ \text{bits}}}$$

5.6.4 Circuit for Dividing Polynomials

We have seen that the cyclic shift of a codeword polynomial and that the encoding of a message polynomial involves the division of one polynomial by another. Such an operation is readily accomplished by a *dividing circuit* (feedback shift register). Given two polynomials $\mathbf{V}(X)$ and $\mathbf{g}(X)$, where

$$\mathbf{V}(X) = v_0 + v_1X + v_2X^2 + \cdots + v_mX^m$$

and

$$\mathbf{g}(X) = g_0 + g_1X + g_2X^2 + \cdots + g_rX^r$$

such that $m \geq r$, the divider circuit of Figure 5.17 performs the polynomial division steps of dividing $\mathbf{V}(X)$ by $\mathbf{g}(X)$, thereby determining the quotient and remainder terms:

$$\frac{\mathbf{V}(X)}{\mathbf{g}(X)} = \mathbf{q}(X) + \frac{\mathbf{r}(X)}{\mathbf{g}(X)}$$

The stages of the register are first initialized by being filled with zeros. The first r shifts enter the most significant (higher-order) coefficients of $\mathbf{V}(X)$. After the rth shift, the quotient output is $g_r^{-1}v_m$; this is the highest-order term in the quotient. For each quotient coefficient q_i the polynomial $q_i\mathbf{g}(X)$ must be subtracted from the dividend. The feedback connections in Figure 5.17 perform this subtraction. The difference between the leftmost r terms remaining in the dividend

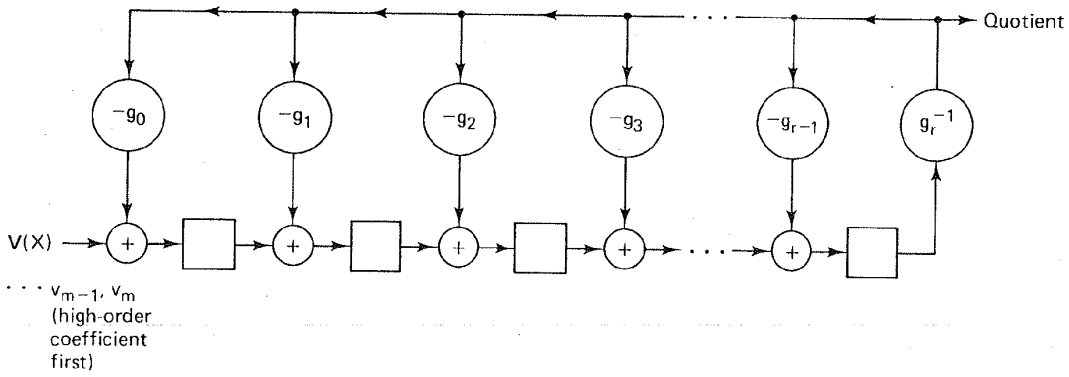


Figure 5.17 Circuit for dividing polynomials.

and the feedback terms $q_i\mathbf{g}(X)$ is formed on each shift of the circuit and appears as the contents of the register. At each shift of the register, the difference is shifted one stage; the highest-order term (which by construction is zero) is shifted out, while the next significant coefficient of $\mathbf{V}(X)$ is shifted in. After $m + 1$ total shifts into the register, the quotient has been serially presented at the output and the remainder resides in the register.

Example 5.9 Dividing Circuit

Use a dividing circuit of the form shown in Figure 5.17 to divide $\mathbf{V}(X) = X^3 + X^5 + X^6$ ($\mathbf{V} = 0\ 0\ 0\ 1\ 0\ 1\ 1$) by $\mathbf{g}(X) = (1 + X + X^3)$. Find the quotient and remainder terms. Compare the circuit implementation to the polynomial division steps performed by hand.

Solution

The dividing circuit needs to perform the following operation:

$$\frac{X^3 + X^5 + X^6}{1 + X + X^3} = \mathbf{q}(X) + \frac{\mathbf{r}(X)}{1 + X + X^3}$$

The required feedback shift register, following the general form of Figure 5.17, is shown in Figure 5.18. Assume that the register contents are initially zero. The operational steps of the circuit are as follows:

Input queue	Shift number	Register contents	Output
0 0 0 1 0 1 1	0	0 0 0	–
0 0 0 1 0 1	1	1 0 0	0
0 0 0 1 0	2	1 1 0	0
0 0 0 1	3	0 1 1	0
0 0 0	4	0 1 1	1
0 0	5	1 1 1	1
0	6	1 0 1	1
–	7	1 0 0	1

After the fourth shift, the quotient coefficients $\{q_i\}$ serially presented at the output are seen to be 1 1 1 1, or the quotient polynomial is $\mathbf{q}(X) = 1 + X + X^2 + X^3$. The remainder coefficients $\{r_i\}$ are 1 0 0, or the remainder polynomial $\mathbf{r}(X) = 1$. In

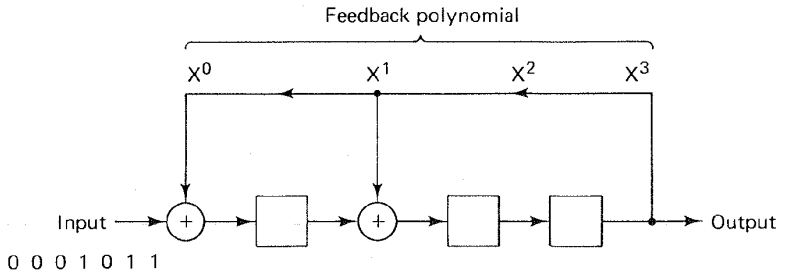


Figure 5.18 Dividing circuit for Example 5.9.

summary, the circuit computation $\mathbf{V}(X)/\mathbf{g}(X)$ is seen to be

$$\frac{X^3 + X^5 + X^6}{1 + X + X^3} = 1 + X + X^2 + X^3 + \frac{1}{1 + X + X^3}$$

The polynomial division steps are as follows:

Output after shift number:

4 5 6 7

$\downarrow$ $\downarrow$ $\downarrow$ $\downarrow$

$X^3 + X^2 + X + 1$

$X^3 + X + 1$ $)X^6 + X^5 + X^3$

$X^6 + X^4 + X^3 \leftarrow$ feedback after 4th shift

$X^5 + X^4 \leftarrow$ register after 4th shift

$X^5 + X^3 + X^2 \leftarrow$ feedback after 5th shift

$X^4 + X^3 + X^2 \leftarrow$ register after 5th shift

$X^4 + X^2 + X \leftarrow$ feedback after 6th shift

$X^3 + X \leftarrow$ register after 6th shift

$X^3 + X + 1 \leftarrow$ feedback after 7th shift

$1 \leftarrow$ register after 7th shift (remainder)

5.6.5 Systematic Encoding with an $(n - k)$-Stage Shift Register

The encoding of a cyclic code in systematic form has been shown, in Section 5.6.3, to involve the computation of parity bits as the result of the formation of $X^{n-k}\mathbf{m}(X)$ modulo $\mathbf{g}(X)$, in other words, the *division* of an *upshifted* (right shifted) message polynomial by a generator polynomial $\mathbf{g}(X)$. The need for upshifting is to make room for the parity bits, which are appended to the message bits, yielding the code vector in systematic form. Upshifting the message bits by $n - k$ positions is a trivial operation and is not really performed as part of the dividing circuit. Instead, only the parity bits are computed; they are then placed in the appropriate location alongside the message bits. The parity polynomial is the *remainder* after dividing by the generator polynomial; it is available in the register after n shifts through the $(n - k)$-stage feedback register shown in Figure 5.18. Notice that the first $n - k$ shifts through the register are simply filling the register. We cannot have any feedback until the rightmost stage has been filled; we therefore can shorten the shifting cycle by loading the input data to the output of the last stage, as shown in Figure 5.19. Further, the feedback term into the leftmost stage is the sum of the input and the rightmost stage. We guarantee that this sum is generated by ensuring that $g_0 = g_{n-k} = 1$ for any generator polynomial $\mathbf{g}(X)$. The circuit feedback connections correspond to the coefficients of the generator polynomial, which is written

$$\mathbf{g}(X) = 1 + g_1 X + g_2 X^2 + \cdots + g_{n-k-1} X^{n-k-1} + X^{n-k} \qquad (5.66)$$

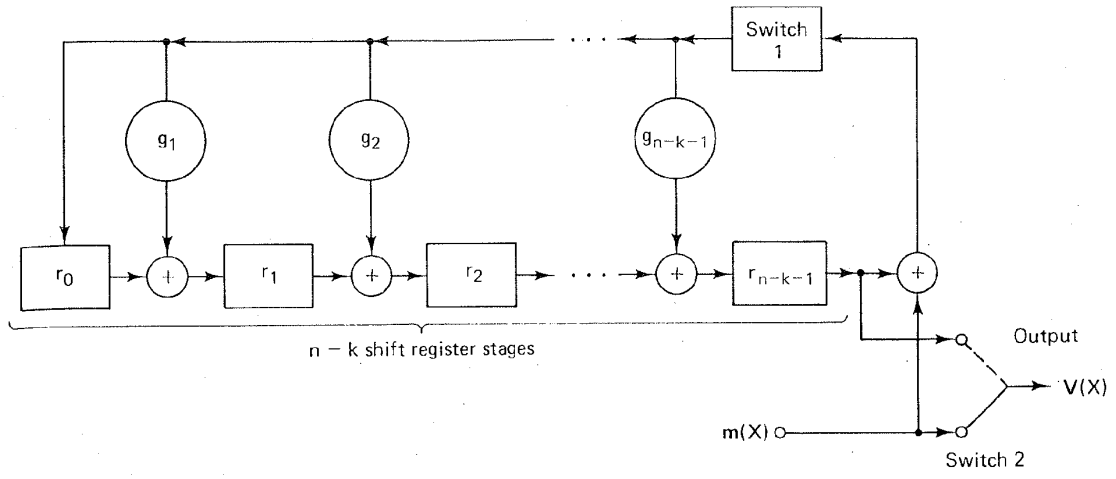


Figure 5.19 Encoding with an $(n - k)$-stage shift register.

The following steps describe the encoding procedure used with the Figure 5.19 encoder.

1. Switch 1 is closed during the first k shifts, to allow transmission of the message bits into the $n - k$ stage encoding shift register.
2. Switch 2 is in the down position to allow transmission of the message bits directly to an output register during the first k shifts.
3. After transmission of the kth message bit, switch 1 is opened and switch 2 is moved to the up position.
4. The remaining $n - k$ shifts clear the encoding register by moving the parity bits to the output register.
5. The total number of shifts is equal to n, and the contents of the output register is the codeword polynomial $\mathbf{r}(X) + X^{n-k}\mathbf{m}(X)$.

Example 5.10 Systematic Encoding of a Cyclic Code

Use a feedback shift register of the form shown in Figure 5.19 to encode the message vector $\mathbf{m} = 1\ 0\ 1\ 1$ into a (7, 4) code vector using the generator polynomial $\mathbf{g}(X) = 1 + X + X^3$.

Solution

$$\mathbf{m} = 1\ 0\ 1\ 1$$

$$\mathbf{m}(X) = 1 + X^2 + X^3$$

$$X^{n-k}\mathbf{m}(X) = X^3\mathbf{m}(X) = X^3 + X^5 + X^6$$

$$X^{n-k}\mathbf{m}(X) = \mathbf{q}(X)\mathbf{g}(X) + \mathbf{r}(X)$$

$$\mathbf{r}(X) = X^3 + X^5 + X^6 \text{ modulo } (1 + X + X^3)$$

For the $(n - k) = 3$-stage encoding shift register shown in Figure 5.20, the operational steps are as follows:

Input queue	Shift number	Register contents	Output
1 0 1 1	0	0 0 0	–
1 0 1	1	1 1 0	1
1 0	2	1 0 1	1
1	3	1 0 0	0
–	4	1 0 0	1

After the fourth shift, switch 1 is opened, switch 2 is moved to the up position, and the parity bits contained in the register are shifted to the output. The output code vector is $\mathbf{U} = 1\ 0\ 0\ 1\ 0\ 1\ 1$, or in polynomial form, $\mathbf{U}(X) = 1 + X^3 + X^5 + X^6$.

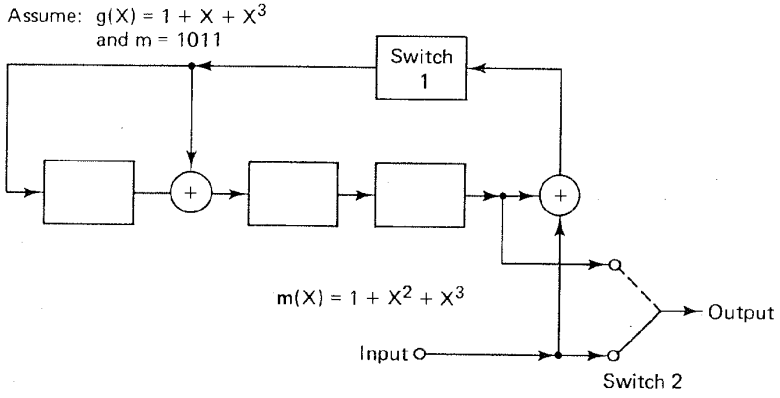


Figure 5.20 Example of encoding a (7, 4) cyclic code with an $(n - k)$-stage shift register.

5.6.6 Error Detection with an $(n - k)$-Stage Shift Register

A transmitted code vector may be perturbed by noise, and hence the vector received may be a corrupted version of the transmitted code vector. Let us assume that a codeword with polynomial representation $\mathbf{U}(X)$ is transmitted and that a vector with polynomial representation $\mathbf{Z}(X)$ is received. Since $\mathbf{U}(X)$ is a code polynomial, it must be a multiple of the generator polynomial $\mathbf{g}(X)$, that is,

$$\mathbf{U}(X) = \mathbf{m}(X)\mathbf{g}(X) \tag{5.67}$$

and $\mathbf{Z}(X)$, the corrupted version of $\mathbf{U}(X)$, can be written

$$\mathbf{Z}(X) = \mathbf{U}(X) + \mathbf{e}(X) \tag{5.68}$$

where $\mathbf{e}(X)$ is the error pattern polynomial. The decoder tests whether $\mathbf{Z}(X)$ is a codeword polynomial, that is, whether it is divisible by $\mathbf{g}(X)$, with a zero re-

mainder. This is accomplished by *calculating the syndrome* of the received polynomial. The syndrome $\mathbf{S}(X)$ is equal to the remainder resulting from dividing $\mathbf{Z}(X)$ by $\mathbf{g}(X)$, that is,

$$\mathbf{Z}(X) = \mathbf{q}(X)\mathbf{g}(X) + \mathbf{S}(X) \tag{5.69}$$

where $\mathbf{S}(X)$ is a polynomial of degree $n - k - 1$ or less. Thus the syndrome is an $(n - k)$-tuple. By combining Equations (5.67) to (5.69), we obtain

$$\mathbf{e}(X) = [\mathbf{m}(X) + \mathbf{q}(X)]\mathbf{g}(X) + \mathbf{S}(X) \tag{5.70}$$

By comparing Equations (5.69) and (5.70), we see that the syndrome $\mathbf{S}(X)$, obtained as the remainder of $\mathbf{Z}(X)$ modulo $\mathbf{g}(X)$, is exactly the same polynomial obtained as the remainder of $\mathbf{e}(X)$ modulo $\mathbf{g}(X)$. Thus the syndrome of the received polynomial $\mathbf{Z}(X)$ contains the information needed for correction of the error pattern. The syndrome calculation is accomplished by a division circuit, almost identical to the encoding circuit used at the transmitter. An example of syndrome calculation with an $n - k$ shift register is shown in Figure 5.21 using the code vector generated in Example 5.10. Switch 1 is initially closed, and switch 2 is open. The received vector is shifted into the register input, with all stages initially set to zero. After the entire received vector has been entered into the shift register, the contents of the register is the syndrome. Switch 1 is then opened and switch 2 is closed, so that the syndrome vector can be shifted out of the register. The operational steps of the decoder are as follows:

Input queue	Shift number	Register contents	
1 0 0 1 0 1 1	0	0 0 0	
1 0 0 1 0 1	1	1 0 0	
1 0 0 1 0	2	1 1 0	
1 0 0 1	3	0 1 1	
1 0 0	4	0 1 1	
1 0	5	1 1 1	
1	6	1 0 1	
–	7	0 0 0	Syndrome

If the syndrome is an all-zeros vector, the received vector is assumed to be a valid code vector. If the syndrome is a nonzero vector, the received vector is a

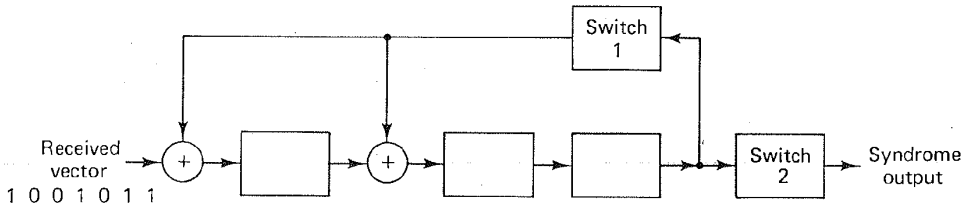


Figure 5.21 Example of syndrome calculation with an $(n - k)$-stage shift register.

perturbed code vector and errors have been detected; such errors can be corrected by adding the error vector (indicated by the syndrome) to the received vector, similar to the procedure described in Section 5.4.8. This method of decoding is useful for simple codes. More complex codes require the use of algebraic techniques to obtain practical decoders [6, 7].

5.7 WELL-KNOWN BLOCK CODES

5.7.1 Hamming Codes

Hamming codes are a simple class of block codes characterized by the following (n, k) structure:

$$(n, k) = (2^m - 1, 2^m - 1 - m) \tag{5.71}$$

where $m = 2, 3, \ldots$. These codes have a minimum distance of 3 and thus, from Equations (5.44) and (5.47), they are capable of correcting all single errors or detecting all combinations of two or fewer errors within a block. Syndrome decoding is especially suited for Hamming codes. In fact, the syndrome can be formed to act as a binary pointer to identify the error location [5]. Although Hamming codes are not very powerful, they belong to a very limited class of block codes known as *perfect* codes, described in Section 5.5.4.

Assuming hard decision decoding the bit error probability can be written, from Equation (5.46), as follows:

$$P_B \simeq \frac{1}{n} \sum_{j=2}^{n} j \binom{n}{j} p^j (1 - p)^{n-j} \tag{5.72}$$

where p is the channel symbol error probability (transition probability on the binary symmetric channel). In place of Equation (5.72) we can use the following equivalent equation. Its identity with Equation (5.72) is proven in Appendix D, Equation (D.16).

$$P_B \cong p - p(1 - p)^{n-1} \tag{5.73}$$

Figure 5.22 is a plot of P_B versus channel symbol error probability, illustrating the comparative performance for different types of block codes. For the Hamming codes, the plots are shown for $m = 3$, 4, and 5, or $(n, k) = (7, 4)$, (15, 11), and (31, 26). For performance over a Gaussian channel using coherently demodulated BPSK, we can express the channel symbol error probability in terms of E_c/N_0, similar to Equation (3.84), as follows:

$$p = Q\left(\sqrt{\frac{2E_c}{N_0}}\right) \tag{5.74}$$

where E_c/N_0 is the code symbol energy per noise spectral density, and where $Q(x)$ is as defined in Equation (2.42). To relate E_c/N_0 to information bit energy per noise spectral density (E_b/N_0), we use

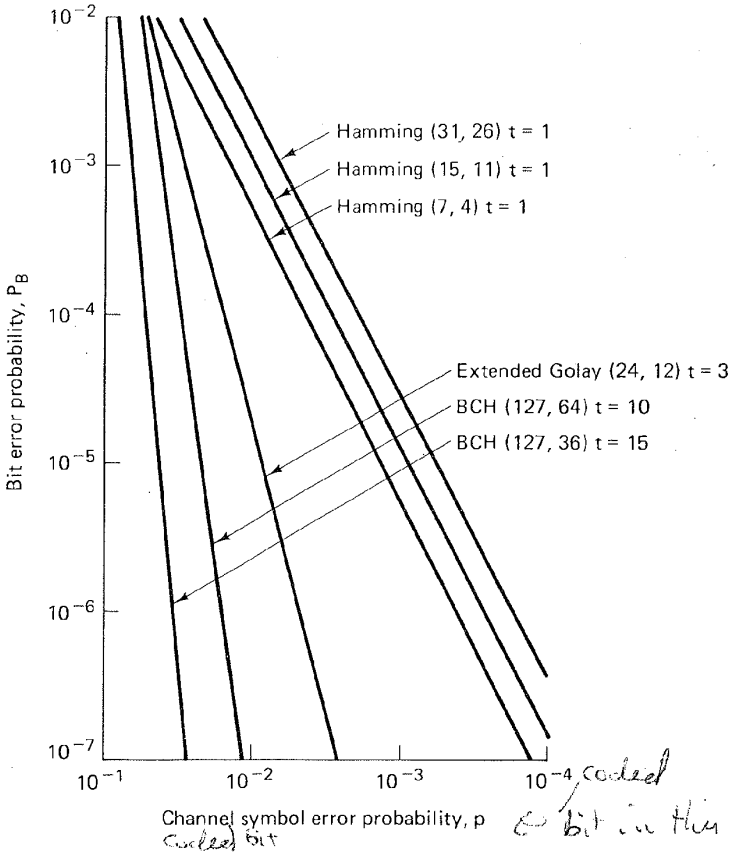


Figure 5.22 Bit error probability versus channel symbol error probability for several block codes.

$$\frac{E_c}{N_0} = \left(\frac{k}{n}\right)\frac{E_b}{N_0} \tag{5.75}$$

For Hamming codes, Equation (5.75) becomes

$$\frac{E_c}{N_0} = \frac{2^m - 1 - m}{2^m - 1}\frac{E_b}{N_0} \tag{5.76}$$

Combining Equations (5.73), (5.74), and (5.76), P_B can be expressed as a function of E_b/N_0 for coherently demodulated BPSK over a Gaussian channel. The results are plotted in Figure 5.23 for different types of block codes. For the Hamming codes, plots are shown for $(n, k) = (7, 4)$, (15, 11), and (31, 26).

Example 5.11 Error Probability for Modulated and Coded Signals

A coded BFSK modulated signal is transmitted over a Gaussian channel. The signal is noncoherently detected and hard-decision decoded. Find the decoded bit error probability if the coding is a Hamming (7, 4) block code and the received E_b/N_0 is equal to 20.

Solution

First we need to find E_c/N_0 using Equation (5.75):

$$\frac{E_c}{N_0} = \frac{4}{7}(20) = 11.43$$

Then, for coded noncoherent BFSK, we can relate the probability of a channel symbol error to E_c/N_0, similar to Equation (3.111), as follows:

$$p = \frac{1}{2}\exp\left(-\frac{E_c}{2N_0}\right)$$

$$= \frac{1}{2}\exp\left(-\frac{11.43}{2}\right) = 1.6 \times 10^{-3}$$

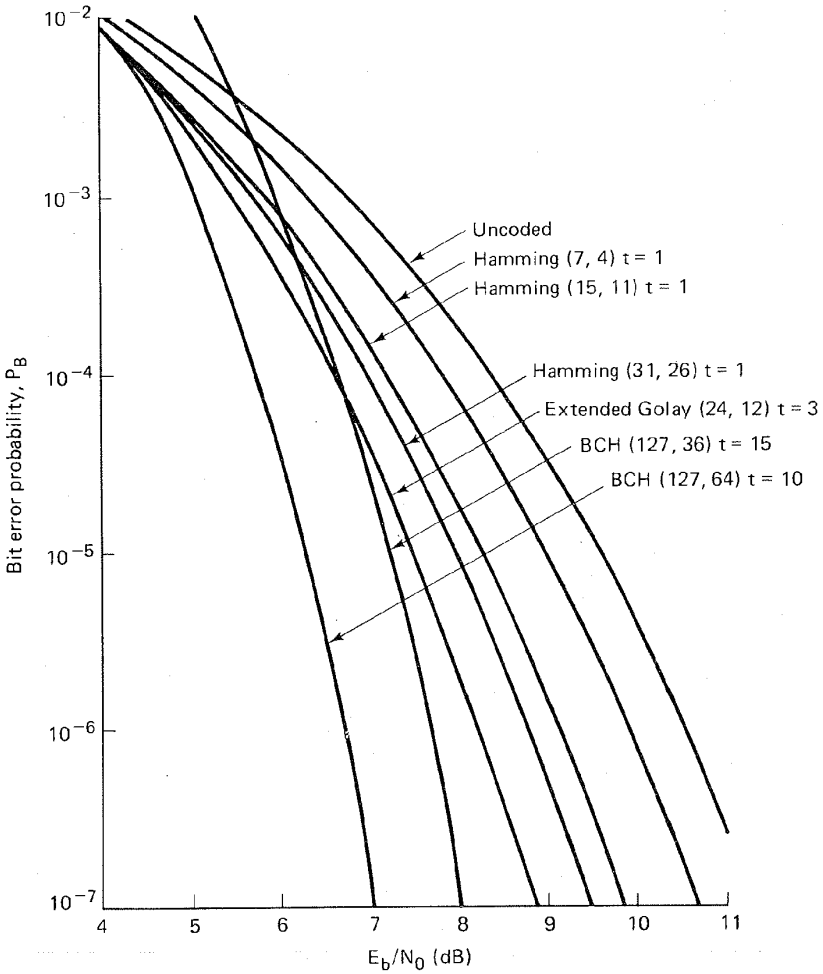


Figure 5.23 P_B versus E_b/N_0 for coherently demodulated BPSK over a Gaussian channel for several block codes.

Using this result in Equation (5.73), we solve for the probability of a decoded bit error, as follows:

$$P_B \simeq p - p(1 - p)^6 \simeq 1.6 \times 10^{-5}$$

5.7.2 Extended Golay Code

One of the more useful block codes is the binary (24, 12) *extended Golay code*, which is formed by adding an overall parity bit to the perfect (23, 12) code, known as the *Golay code*. This added parity bit increases the minimum distance $d_{\min}$ from 7 to 8 and produces a rate $\frac{1}{2}$ code, which is easier to implement (with regard to system clocks) than the rate 12/23 original Golay code. Extended Golay codes are considerably more powerful than the Hamming codes described in the preceding section. The price paid for the improved performance is a more complex decoder, a lower code rate, and hence a larger bandwidth expansion.

Since $d_{\min} = 8$ for the extended Golay code, we see from Equation (5.44) that the code is guaranteed to correct all triple errors. The decoder can additionally be designed to correct *some but not all* four-error patterns. Since only 19% of the four-error patterns can be corrected, the decoder, for the sake of simplicity, is usually designed to only correct three-error patterns [5]. Assuming hard decision decoding, the bit error probability for the extended Golay code can be written as a function of the channel symbol error probability, p, from Equation (5.46), as follows:

$$P_B \simeq \frac{1}{24} \sum_{j=4}^{24} j \binom{24}{j} p^j (1 - p)^{24-j} \tag{5.77}$$

The plot of Equation (5.77) is shown in Figure 5.22; the error performance of the extended Golay code is seen to be significantly better than that of the Hamming codes. Combining Equations (5.77), (5.74), and (5.75), we can relate P_B versus E_b/N_0 for coherently demodulated BPSK with extended Golay coding over a Gaussian channel. The result is plotted in Figure 5.23.

5.7.3 BCH Codes

Bose–Chadhuri–Hocquenghem (BCH) codes are a generalization of Hamming codes that allow multiple error correction. They are a *powerful class of cyclic codes* that provide a large selection of block lengths, code rates, alphabet sizes, and error-correcting capability. Table 5.2 lists some commonly used code generators, $\mathbf{g}(x)$, for the construction of BCH codes [8], for various values of n, k, and t, up to a block length of 255. The coefficients of $\mathbf{g}(x)$ are presented as octal numbers arranged so that when they are converted to binary digits the rightmost digit corresponds to the zero-degree coefficient of $\mathbf{g}(x)$. BCH codes are important, because at block lengths of a few hundred, the BCH codes outperform all other block codes with the same block length and code rate. The most commonly used BCH codes employ a binary alphabet and a codeword block length of $n = 2^m - 1$, where $m = 3, 4, \ldots$.

TABLE 5.2 Generators of Primitive BCH Codes

n	k	t	g(x)	n	k	t	g(x)
7	4	1	13	255	171	11	1541621421234235607706163063 7
15	11	1	23		163	12	75004155100756025515747245146 01
	7	2	721		155	13	37575130054076650157225064646776 33
	5	3	2467		147	14	16421301735371655253041653054410117 11
31	26	1	45		139	15	46140173206017556157072273024745356744 5
	21	2	3551		131	18	21571333147151015126125027744214202 41
	16	3	107657				65471
	11	5	5423325		123	19	12061405224206600371721032651614122 62
	6	7	313365047				72506267
63	57	1	103		115	21	60526665572100247263636404600276352 55
	51	2	12471				6313472737
	45	3	1701317		107	22	22205772322066256312417300235347420 17
	39	4	166623567				6574750154441
	36	5	1033500423		99	23	10656667253473174222741416201574332 25
	30	6	157464165547				2411076432303431
	24	7	17323260404441		91	25	67502650303274441727236317247325110 75
	18	10	1363026512351725				550762720724344561
	16	11	6331141367235453		87	26	11013676341474323643523163430717204 62
	10	13	472622305527250155				06722545273311721317
	7	15	5231045543503271737		79	27	66700035637657500020270344207366174 62
							1015326711766541342355

127	120	1	211
	113	2	41567
	106	3	11554743
	99	4	3447023271
	92	5	624730022327
	85	6	130704476322273
	78	7	26230002166130115
	71	9	6255010713253127753
	64	10	1206534025570773100045
	57	11	335265252505705053517721
	50	13	544465125233140124215014 21
	43	14	177217722136512275212205743 43
	36	15	31460746665220750447645747217 35
	29	21	40311446136767060366753014117 6155
	22	23	12337607040472252243544562663764 7043
	15	27	22057042445604554770523013762217604353
	8	31	70472640527510306514762242715677331302 17
255	247	1	435
	239	2	267543
	231	3	156720665
	223	4	75626641375
	215	5	23157564726421
	207	6	16176560567636227
	199	7	7633031270420722341
	191	8	2663470176115333714567
	187	9	52755313540001322236351
	179	10	22624710717340432416300455
	71	29	24024710520644321515554172112331163205444250362557643221706035
	63	30	10754475055163544325315217357707003666111726455267613656702543301
	55	31	73154252035011001330152753060320543254143267550105570444260354736 17
	47	42	25335420170626465630330413774062331751233341454460450050660245525431 73
	45	43	15202056055234161131101346376423701563670024470762373033202157025051541
	37	45	51363302550670074141774472454375304207357061743234323476443547374030440 03
	29	47	30257155366730714655270640123613771153422423242011741140602547574104035650 37
	21	55	12562152570603326560017731536076121032273414056530745425211531216144665134 73725
	13	59	46417320050525645444265737142500660043306774454765614031746772135702613446 0500547
	9	63	15726025217472463201031043255355134614162367212044074545112766115547705561677516057

Source: Reprinted with permission from "Table of Generators for BCH Codes," *IEEE Trans. Inf. Theory*, vol. IT10, no. 4, Oct. 1964, p. 391. © 1964 IEEE.

The title of Table 5.2 indicates that the generators shown are for those BCH codes known as *primitive codes*. The term "primitive" is a number-theoretic concept requiring an elaborate algebraic development [9–11], which will not be presented here. In Figures 5.22 and 5.23 are plotted error performance curves of two BCH codes (127, 64) and (127, 36), to illustrate comparative performance. Assuming hard decision decoding, the P_B versus channel error probability is shown in Figure 5.22. The P_B versus E_b/N_0 for coherently demodulated BPSK over a Gaussian channel is shown in Figure 5.23. The curves in Figure 5.23 seem to depart from our expectations. They each have the same block size, yet the more redundant (127, 36) code does not exhibit as much coding gain as does the less redundant (127, 64) code. It has been shown that a relatively broad maximum of coding gain versus code rate for fixed n occurs roughly between coding rates of $\frac{1}{3}$ and $\frac{3}{4}$ for BCH codes [12]. Performance over a Gaussian channel degrades substantially at very high or very low rates [11].

Figure 5.24 represents computed performance of BCH codes [13] using coherently demodulated BPSK with both *hard-* and *soft-decision decoding*. Soft-decision decoding is not usually used with block codes because of its complexity. However, whenever it is implemented, it offers an approximate 2-dB coding gain over hard-decision decoding. For a given code rate, the decoded error probability is known to improve with increasing block length n [4]. Thus for a given code rate, it is interesting to compare the block length that would be required for the hard-decision-decoding performance to be comparable to the soft-decision-decoding performance. In Figure 5.24, the BCH codes shown all have code rates of approximately $\frac{1}{2}$. From the figure [13] it appears that for a fixed code rate, the hard-decision-decoded BCH code of length 8 times n or longer has a better performance than that of a soft-decision-decoded BCH code of length n.

5.7.4 Reed–Solomon Codes

One special subclass of the BCH codes (the discovery of which preceded the BCH codes) is the particularly useful *nonbinary* set called Reed–Solomon codes. Reed–Solomon codes achieve the *largest possible code minimum distance* for any linear code with the same encoder input and output block lengths. For nonbinary codes, the distance between two code words is defined as the number of nonbinary symbols in which the sequences differ. For Reed–Solomon codes the code minimum distance is given by [14]

$$d_{\min} = n - k + 1 \tag{5.78}$$

where k is the number of data symbols being encoded, and n is the total number of code symbols in the encoded block. Following Equation (5.44), the code is capable of correcting any combination of t or fewer symbol errors, as follows:

$$t = \frac{d_{\min} - 1}{2} = \frac{n - k}{2} \tag{5.79}$$

and thus requires no more than $2t$ parity check symbols.

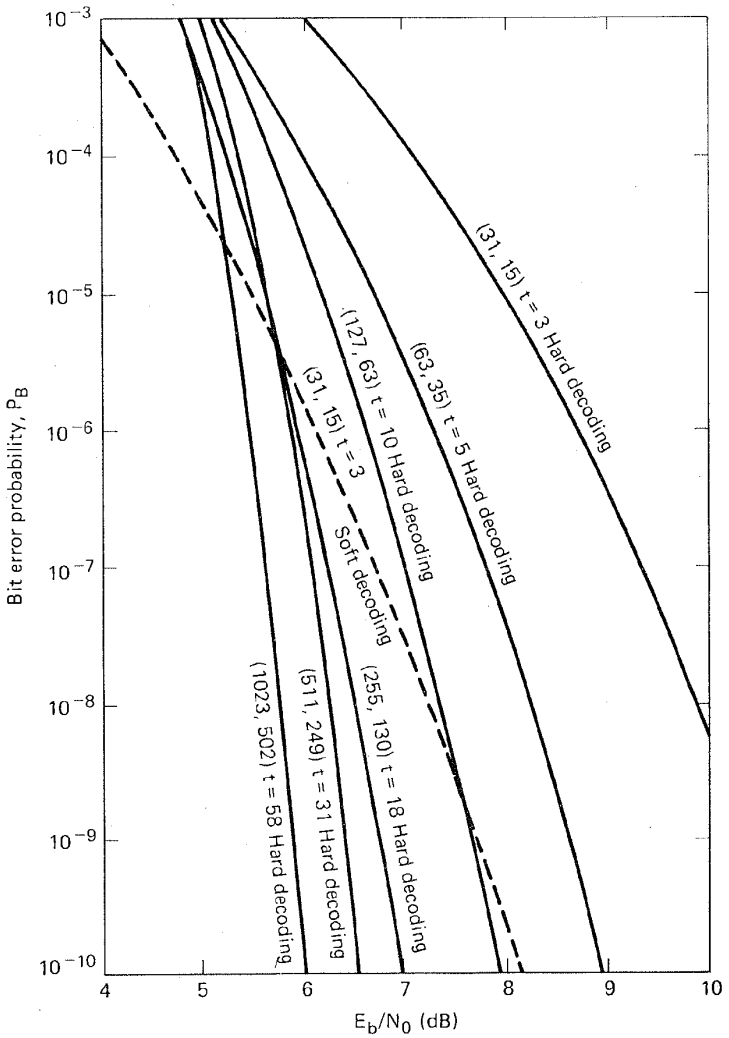


Figure 5.24 P_B versus E_b/N_0 for coherently demodulated BPSK over a Gaussian channel using BCH codes. (Reprinted with permission from L. J. Weng, "Soft and Hard Decoding Performance Comparisons for BCH Codes," *Proc. Int. Conf. Commun.*, 1979, Fig. 3, p. 25.5.5. © 1979 IEEE.)

A t-error-correcting Reed–Solomon code with an alphabet of 2^m symbols has $n = 2^m - 1$ and $k = 2^m - 1 - 2t$, where $m = 2, 3, \ldots$. An advantage of nonbinary codes such as a Reed–Solomon code can be seen by the following comparison. Consider a binary $(n, k) = (7, 3)$ code. The entire n-tuple space amounts to $2^n = 2^7 = 128$ binary words, of which $2^k = 2^3 = 8$ (or $\frac{1}{16}$ of the n-tuples) are codewords. Next, consider a nonbinary $(n, k) = (7, 3)$ code where each symbol is comprised of $m = 3$ bits. The n-tuple space amounts to $2^{nm} = 2^{21} = 2{,}097{,}152$ binary words, of which $2^{km} = 2^9 = 512$ (or 1/4096 of the n-tuples) are codewords. With symbols, each made up of m bits, only a small fraction (i.e., 2^{km} of the large number 2^{nm}) of possible different words of n symbols become

codewords. This fraction decreases with increasing values of m. When a small fraction of the n-tuple space is used for codewords, a large $d_{\min}$ can be created.

The Reed–Solomon (R–S) codes are particularly useful for *burst-error correction*; that is, they are effective for channels that have memory. Also, they can be used efficiently on channels where the set of input symbols is large. An interesting feature of the R–S code is that as many as two information symbols can be added to an R–S code of length n without reducing its minimum distance. This extended R–S code has length $n + 2$ and the same number of parity check symbols as the original code. From Equation (5.46) the R–S decoded symbol error probability, P_E, can be written in terms of the channel symbol error probability, p, as follows [5]:

$$P_E \simeq \frac{1}{2^m - 1} \sum_{j=t+1}^{2^m-1} j \binom{2^m - 1}{j} p^j (1 - p)^{2^m - 1 - j} \tag{5.80}$$

The bit error probability can be upper bounded by the symbol error probability

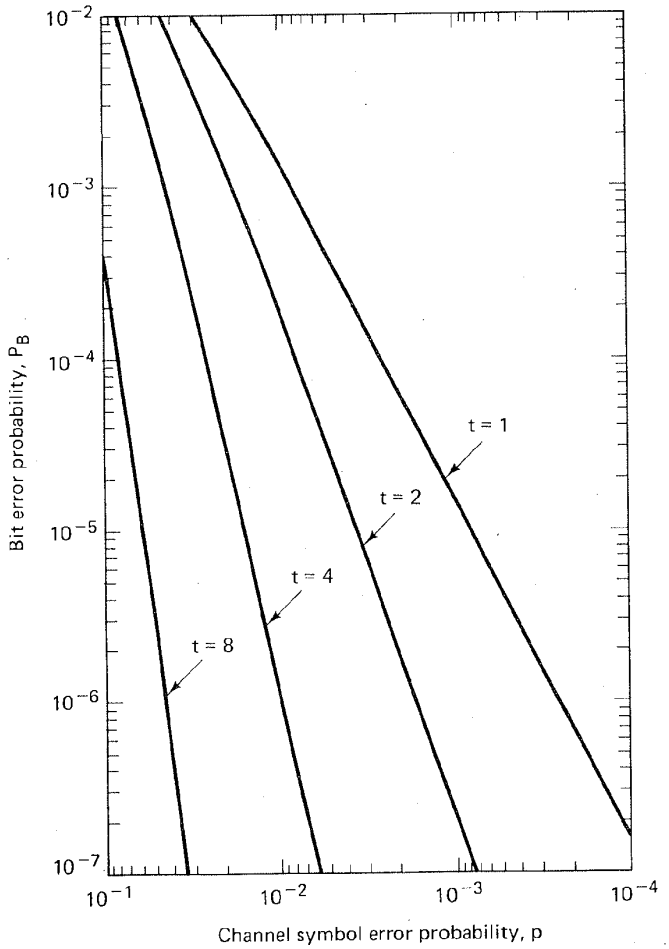


Figure 5.25 P_B versus p for 32-ary orthogonal signaling and $n = 31$, t-error-correcting Reed–Solomon coding. (Reprinted with permission from *Data Communications, Networks, and Systems*, ed. Thomas C. Bartee, Howard W. Sams Company, Indianapolis, Ind., 1985, p. 311. Originally published in J. P. Odenwalder, *Error Control Coding Handbook*, M/A-COM LINKABIT, Inc., San Diego, Calif., July 15, 1976, p. 91.)

for specific modulation types. For MFSK modulation with $M = 2^m$, the relationship between P_B and P_E as given in Equation (3.127) is repeated here:

$$\frac{P_B}{P_E} = \frac{2^{m-1}}{2^m - 1} \tag{5.81}$$

Figure 5.25 shows P_B versus the channel symbol error probability, p, plotted from Equations (5.80) and (5.81) for various t-error-correcting 32-ary orthogonal Reed–

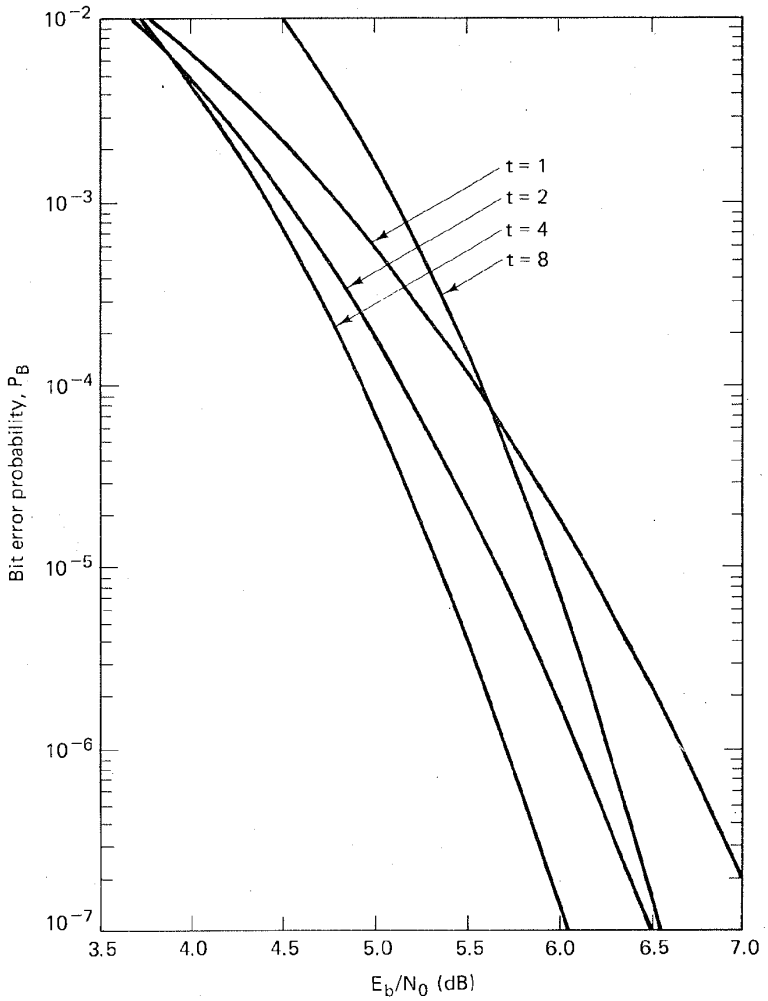


Figure 5.26 Bit error probability versus E_b/N_0 performance of several $n = 31$, t-error-correcting Reed–Solomon coding systems with 32-ary MFSK modulation over an AWGN channel. (Reprinted with permission from *Data Communications, Networks, and Systems*, ed. Thomas C. Bartee, Howard W. Sams Company, Indianapolis, Ind., 1985, p. 312. Originally published in J. P. Odenwalder, *Error Control Coding Handbook*, M/A-COM LINKABIT, Inc., San Diego, Calif., July 15, 1976, p. 92.)

Solomon codes with $n = 31$ (thirty-one 5-bit symbols per code block). Figure 5.26 shows P_B versus E_b/N_0 for such a coded system using 32-ary MFSK modulation and noncoherent demodulation over an AWGN channel [5]. For R–S codes, error probability is an exponentially decreasing function of block length, n, and decoding complexity is proportional to a small power of the block length [14]. The R–S codes are sometimes used in a concatenated arrangement. In such a system, an inner convolutional decoder first provides some error control by operating on soft-decision demodulator outputs; the convolutional decoder then presents hard-decision data to the outer Reed–Solomon decoder, which further reduces the probability of error. In Chapter 6 we discuss further the use of concatenated and R–S coding as applied to the compact disc (CD) digital audio system.

5.8 CONCLUSION

In this chapter we have explored the general goals of channel coding, all leading to improve probability of error performance at a minimum cost in bandwidth. We partitioned channel coding into two study groups: waveform coding and structured sequences. Waveform coding represents a transformation of waveforms into improved waveforms, such that the distance properties are improved compared to the original waveforms. Structured sequences involve the addition of parity digits to the data such that the parity digits can then be employed for detecting and/or correcting specific error patterns. The main advantage of structured sequence coding over waveform coding is that it can accomplish improved P_B performance using less bandwidth.

We particularly examined linear block codes. Geometric analogies can be drawn between the coding and modulation disciplines. They both seek to pack the signal space efficiently and to maximize the distance between signals in the signaling set. Within block codes we looked at cyclic codes, which are relatively easy to implement using modern integrated circuit techniques. We considered the polynomial representation of codes and the correspondence between the polynomial structure, the necessary algebraic operations, and the hardware implementation. We have also looked at performance details of some of the well-known block codes. A large class of codes, the convolutional codes, have been left for consideration in Chapter 6.

REFERENCES

1. Viterbi, A. J., "On Coded Phase-Coherent Communications," *IRE Trans. Space Electron. Telem.*, vol. SET7, Mar. 1961, pp. 3–14.
2. Lindsey, W. C., and Simon, M. K., *Telecommunication Systems Engineering*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1973.

3. Proakis, J. G., *Digital Communications,* McGraw-Hill Book Company, New York, 1983.
4. Lin, S., and Costello, D. J., Jr., *Error Control Coding: Fundamentals and Applications,* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1983.
5. Odenwalder, J. P., *Error Control Coding Handbook,* Linkabit Corporation, San Diego, Calif., July 15, 1976.
6. Blahut, R. E., *Theory and Practice of Error Control Codes,* Addison-Wesley Publishing Company, Inc., Reading, Mass., 1983.
7. Blahut, R. E., "Algebraic Fields, Signal Processing, and Error Control," *Proc. IEEE,* vol. 73, May 1985, pp. 874–893.
8. Stenbit, J. P., "Tables of Generators for Bose–Chadhuri Codes, *IEEE Trans. Inf. Theory,* vol. IT10, no. 4, Oct. 1964, pp. 390–391.
9. Berlekamp, E. R., *Algebraic Coding Theory,* McGraw-Hill Book Company, New York, 1968.
10. Peterson, W. W., and Weldon, E. J., *Error Correcting Codes,* 2nd ed., The MIT Press, Cambridge, Mass., 1972.
11. Clark, G. C., Jr., and Cain, J. B., *Error-Correction Coding for Digital Communications,* Plenum Press, New York, 1981.
12. Wozencraft, J. M., and Jacobs, I. M., *Principles of Communication Engineering,* John Wiley & Sons, Inc., New York, 1965.
13. Weng, L. J., "Soft and Hard Decoding Performance Comparisons for BCH Codes," *Proc. Int. Conf. Commun.,* 1979, pp. 25.5.1–25.5.5.
14. Gallager, R. G., *Information Theory and Reliable Communication,* John Wiley & Sons, Inc., New York, 1968.

PROBLEMS

5.1. Design an (n, k) single-parity code that will detect all 1-, 3-, 5-, and 7-error patterns in a block. Show the values of n and k, and find the probability of an undetected block error if the probability of channel symbol error is 10^{-2}.

5.2. Calculate the probability of message error for a 12-bit data sequence encoded with a (24, 12) linear block code. Assume that the code corrects all 1-bit and 2-bit error patterns and assume that it corrects no error patterns with more than two errors. Also, assume that the probability of a channel symbol error is 10^{-3}.

5.3. Consider a (127, 92) linear block code capable of triple error corrections.
(a) What is the probability of message error for an uncoded block of 92 bits if the channel symbol error probability is 10^{-3}?
(b) What is the probability of message error when using the (127, 92) block code if the channel symbol error probability is 10^{-3}?

5.4. Calculate the improvement in probability of message error relative to an uncoded transmission for a (24, 12) double-error-correcting linear block code. Assume that coherent BPSK modulation is used and that the received $E_b/N_0 = 10$ dB.

5.5. Consider a (24, 12) linear block code capable of double-error corrections. Assume

that a noncoherently detected BFSK modulation format is used and that the received $E_b/N_0 = 14$ dB.

(a) Does the code provide any improvement in probability of message error? If it does, how much? If it does not, explain why not.

(b) Repeat part (a) with $E_b/N_0 = 10$ dB.

5.6. The telephone company uses a "best-of-five" encoder for some of its digital data channels. In this system every data bit is repeated five times, and at the receiver, a majority vote decides the value of each data bit. If the uncoded probability of bit error is 10^{-3}, calculate the coded bit error probability when using such a best-of-five code.

5.7. The minimum distance for a particular linear block code is 11. Find the maximum error-correcting capability, the maximum error-detecting capability, and the maximum erasure-correcting capability in a block length.

5.8. Consider a (7, 4) code whose generator matrix is

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

(a) Find all the code vectors of the code.

(b) Find $\mathbf{H}$, the parity-check matrix of the code.

(c) Compute the syndrome for the received vector 1 1 0 1 1 0 1. Is this a valid code vector?

(d) What is the error-correcting capability of the code?

(e) What is the error-detecting capability of the code?

5.9. Consider a systematic block code whose parity-check equations are

$$p_1 = m_1 + m_2 + m_4$$

$$p_2 = m_1 + m_3 + m_4$$

$$p_3 = m_1 + m_2 + m_3$$

$$p_4 = m_2 + m_3 + m_4$$

where m_i are message digits and p_i are check digits.

(a) Find the generator matrix and the parity-check matrix for this code.

(b) How many errors can the code correct?

(c) Is the vector 10101010 a codeword?

(d) Is the vector 01011100 a codeword?

5.10. Consider the linear block code with the codeword defined by

$$\mathbf{U} = m_1 + m_2 + m_4 + m_5,\ m_1 + m_3 + m_4 + m_5,\ m_1 + m_2 + m_3 + m_5,\ m_1 + m_2 + m_3 + m_4,\ m_1,\ m_2,\ m_3,\ m_4,\ m_5$$

(a) Show the generator matrix.

(b) Show the parity-check matrix.

(c) Find n, k, and $d_{\min}$.

5.11. Design a (4, 2) linear block code.
- **(a)** Choose the codewords to be in systematic form, and choose them with the goal of maximizing $d_{\min}$.
- **(b)** Find the generator matrix for the codeword set.
- **(c)** Calculate the parity-check matrix.
- **(d)** Enter the sixteen 4-tuples into a standard array.
- **(e)** What are the error-correcting and error-detecting capabilities of the code?
- **(f)** Make a syndrome table for the correctable error patterns.

5.12. Consider the (5, 1) repetition code, which consists of the two codewords 00000 and 11111, corresponding to messages 0 and 1, respectively. Derive the standard array for this code. Is this a perfect code?

5.13. Design a (3, 1) code that will correct all single-error patterns. Choose the codeword set and show the standard array.

5.14. Is a (7, 3) code a perfect code? Is a (7, 4) code a perfect code? Is a (15, 11) code a perfect code? Justify your answers.

5.15. A (15, 11) linear block code can be defined by the following parity array:

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

- **(a)** Show the parity-check matrix for this code.
- **(b)** List the coset leaders from the standard array. Is this code a perfect code? Justify your answer.
- **(c)** A received vector is $\mathbf{V} = 0\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 1$. Compute the syndrome. Assuming that a single bit error has been made, find the correct codeword.
- **(d)** How many erasures can this code correct? Explain.

5.16. Is it possible that a nonzero error pattern can produce a syndrome of $\mathbf{S} = \mathbf{0}$? If yes, how many such error patterns can give this result for an (n, k) code? Use Figure 5.14 to justify your answer.

5.17. Determine which, if any, of the following polynomials can generate a cyclic code with codeword length $n \leq 7$. Find the (n, k) values of any such codes that can be generated.
- **(a)** $1 + X^3 + X^4$
- **(b)** $1 + X^2 + X^4$
- **(c)** $1 + X + X^3 + X^4$
- **(d)** $1 + X + X^2 + X^4$
- **(e)** $1 + X^3 + X^5$

5.18. Encode the message 1 0 1 in systematic form using polynomial division and the generator $\mathbf{g}(X) = 1 + X + X^2 + X^4$.

5.19. Design a feedback shift register encoder for an (8, 5) cyclic code with a generator $\mathbf{g}(x) = 1 + X + X^2 + X^3$. Use the encoder to find the codeword for the message 1 0 1 0 1 in systematic form.

5.20. In Figure P5.1 the signal is differentially coherent PSK (DPSK), the encoded symbol rate is 10,000 code symbols per second, and the decoder is a single-error-correcting (7, 4) decoder. Is a predetection signal-to-noise spectral density ratio of $P_r/N_0 = 48$ dBW sufficient to provide a probability of message error of 10^{-3} at the output? Justify your answer. Assume that a message block contains 4 data bits and that any single-error pattern in a block length of 7 bits can be corrected.

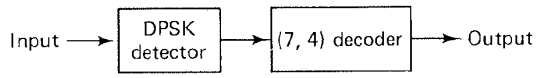


Figure P5.1

5.21. A (15, 5) cyclic code has a generator polynomial as follows:

$$\mathbf{g}(X) = 1 + X + X^2 + X^5 + X^8 + X^{10}$$

(a) Draw a diagram of an encoder for this code.
(b) Find the code polynomial (in systematic form) for the message $\mathbf{m}(X) = 1 + X^2 + X^4$.
(c) Is $\mathbf{V}(X) = 1 + X^4 + X^6 + X^8 + X^{14}$ a code polynomial in this system? Justify your answer.

5.22. Consider the (15, 11) cyclic code generated by $\mathbf{g}(X) = 1 + X + X^4$.
(a) Devise a feedback register encoder and decoder for this code.
(b) Illustrate the encoding procedure with the message vector 11001101011 by listing the states of the register (the rightmost bit is the earliest bit).
(c) Repeat part (b) for the decoding proccdure.

5.23. For a fixed probability of channel symbol error, the probability of bit error for a Hamming (15, 11) code is worse than that for a Hamming (7, 4) code. Explain why. What, then, is the advantage of the (15, 11) code? What basic trade-off is involved?

5.24. A (63, 36) BCH code can correct five errors. Nine blocks of a (7, 4) code can correct nine errors. Both codes have the same code rate.
(a) The (7, 4) code can correct more errors. Is it more powerful? Explain.
(b) Compare the two codes when five errors occur randomly in 63 bits.

5.25. Information from a source is organized in 36-bit messages that are to be transmitted over an AWGN channel using noncoherently detected BFSK modulation.
(a) If no error control coding is used, compute the E_b/N_0 required to provide a message error probability of 10^{-3}.
(b) Consider the use of a (127, 36) linear block code (minimum distance is 31) in the transmission of these messages. Compute the coding gain for this code for a message error probability of 10^{-3}. (*Hint:* The coding gain is defined as the difference between the E_b/N_0 required without coding and the E_b/N_0 required with coding.)

5.26. **(a)** Consider a data sequence encoded with a (127, 64) BCH code and then modulated using coherent 16-ary PSK. If the received E_b/N_0 is 10 dB, find the MPSK probability of symbol error, the probability of coded bit error (assuming that a Gray

code is used for symbol-to-bit assignment), and the probability of information bit error.

(b) For the same probability of information bit error found in part (a), determine the value of E_b/N_0 required if the modulation in part (a) is changed to coherent 16-ary FSK. Explain the difference.

5.27. A message consists of English text (assume that each word in the message contains six letters). Each letter is encoded using the 7-bit ASCII character code. Thus, each word of text consists of a 42-bit sequence. The message is to be transmitted over a channel having a symbol error probability of 10^{-3}.

(a) What is the probability that a word will be received in error?

(b) If a repetition code is used such that each letter in each word is repeated three times, and at the receiver, majority voting is used to decode the message, what is the probability that a decoded word will be in error?

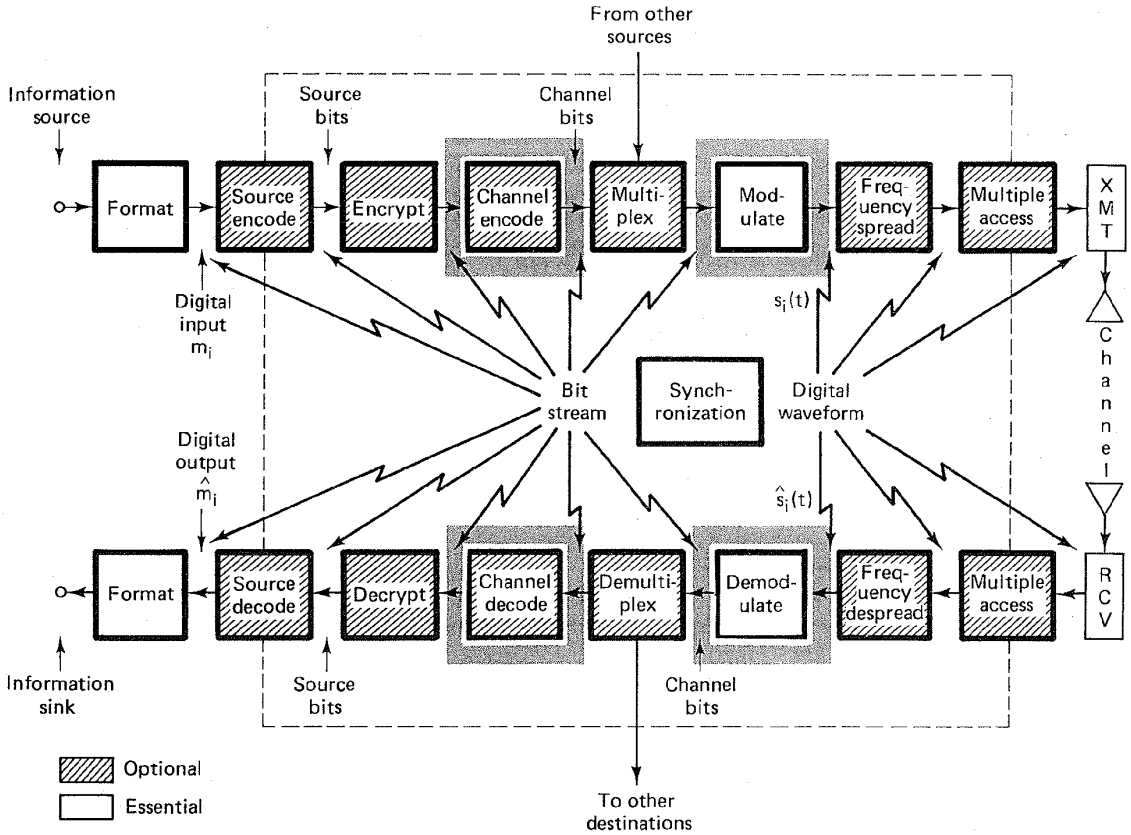
(c) If a (126, 42) BCH code with error-correcting capability of $t = 14$ is used to encode each 42-bit word, what is the probability that a decoded word will be in error?

(d) For a real system, it is not fair to compare uncoded versus coded message error performance on the basis of a fixed probability of channel symbol error, since this implies a fixed level of received E_c/N_0 for all choices of coding (or lack of coding). Therefore, repeat parts (a), (b), and (c) under the condition that the channel symbol error probability is determined by a received E_b/N_0 of 12 dB, where E_b/N_0 is the information bit energy per noise spectral density. Assume that the information rate must be the same for all choices of coding or lack of coding. Also assume that noncoherent binary FSK modulation is used over an AWGN channel.

(e) Discuss the relative error performance capabilities of the above coding schemes under the two postulated conditions—fixed channel symbol error probability, and fixed E_b/N_0. Under what circumstances can a repetition code offer error performance improvement? When will it cause performance degradation?

CHAPTER 7

Modulation and Coding Trade-Offs



7.1 GOALS OF THE COMMUNICATIONS SYSTEM DESIGNER

System trade-offs are fundamental to all digital communication designs. The goals of the designer are (1) to maximize transmission bit rate, R; (2) to minimize probability of bit error, P_B; (3) to minimize required power, or equivalently, to minimize required bit energy to noise power spectral density, E_b/N_0; (4) to minimize required system bandwidth, W; (5) to maximize system utilization, that is, to provide reliable service for a maximum number of users with minimum delay and with maximum resistance to interference; and (6) to minimize system complexity, computational load, and system cost. A good system designer seeks to achieve all these goals simultaneously. However, goals 1 and 2 are clearly in conflict with goals 3 and 4; they call for simultaneously maximizing R, while minimizing P_B, E_b/N_0, and W. There are several constraints and theoretical limitations that necessitate the trading off of any one system requirement with each of the others. Some of the constraints are:

The Nyquist theoretical minimum bandwidth requirement
The Shannon–Hartley capacity theorem (and the Shannon limit)
Government regulations (e.g., frequency allocations)
Technological limitations (e.g., state-of-the art components)
Other system requirements (e.g., satellite orbits)

Some of the realizable modulation and coding trade-offs can best be viewed

as a change in operating point on one of two performance planes. These planes will be referred to as the error probability plane and the bandwidth efficiency plane; they are described in the following sections.

7.2 ERROR PROBABILITY PLANE

Figure 7.1 illustrates the family of P_B versus E_b/N_0 curves for the coherent detection of orthogonal signaling (Figure 7.1a) and multiple phase signaling (Figure 7.1b). For signaling schemes that process k bits at a time, the signaling is called M-ary (see Section 3.8). The modulator uses one of its $M = 2^k$ waveforms to represent each k-bit sequence, where M is the size of the symbol set. Figure 7.1a illustrates the potential bit error improvement with orthogonal signaling as k (or M) is increased. For orthogonal signal sets, such as frequency shift keying (FSK) modulation, increasing the size of the symbol set can provide an improvement in P_B, or a reduction in the E_b/N_0 required, at the cost of increased bandwidth. Figure 7.1b illustrates potential bit error degradation with nonorthogonal signaling as k (or M) increases. For nonorthogonal signal sets, such as multiple phase shift keying (MPSK) modulation, increasing the size of the symbol set can reduce the bandwidth requirement, but at the cost of a degraded P_B, or an increased E_b/N_0 requirement. We shall refer to these families of curves (Figure 7.1a or b) as *error probability performance curves,* and to the plane on which they are plotted as an *error probability plane*. Such a plane describes the locus of operating points available for a particular type of modulation and coding. For a given system information rate, each curve in the plane can be associated with a different fixed minimum required bandwidth; therefore, the set of curves can be termed *equibandwidth curves*. As the curves move in the direction of the ordinate, the required transmission bandwidth increases; as the curves move in the opposite direction, the required bandwidth decreases. Once a modulation and coding scheme and an available E_b/N_0 are determined, system operation is characterized by a particular point in the error probability plane. Possible trade-offs can be viewed as changes in the operating point on one of the curves or as changes in the operating point from one curve to another curve of the family. These trade-offs are seen in Figure 7.1a and b as changes in the system operating point in the direction shown by the arrows. Movement of the operating point along line 1, between points a and b, can be viewed as trading off P_B for E_b/N_0 performance (with W fixed). Similarly, movement along line 2, between points c and d, is seen as trading P_B for W performance (with E_b/N_0 fixed). Finally, movement along line 3, between points e and f, illustrates trading W for E_b/N_0 performance (with P_B fixed). Movement along line 1 is effected by increasing or decreasing the available E_b/N_0. This can be achieved, for example, by increasing transmitter power, which means that the trade-off might be accomplished simply by "turning a knob," even after the system is configured. However, the other trade-offs (movement along line 2 or line 3) involve some change in the system modulation or coding scheme, and therefore need to be accomplished during the system design phase.

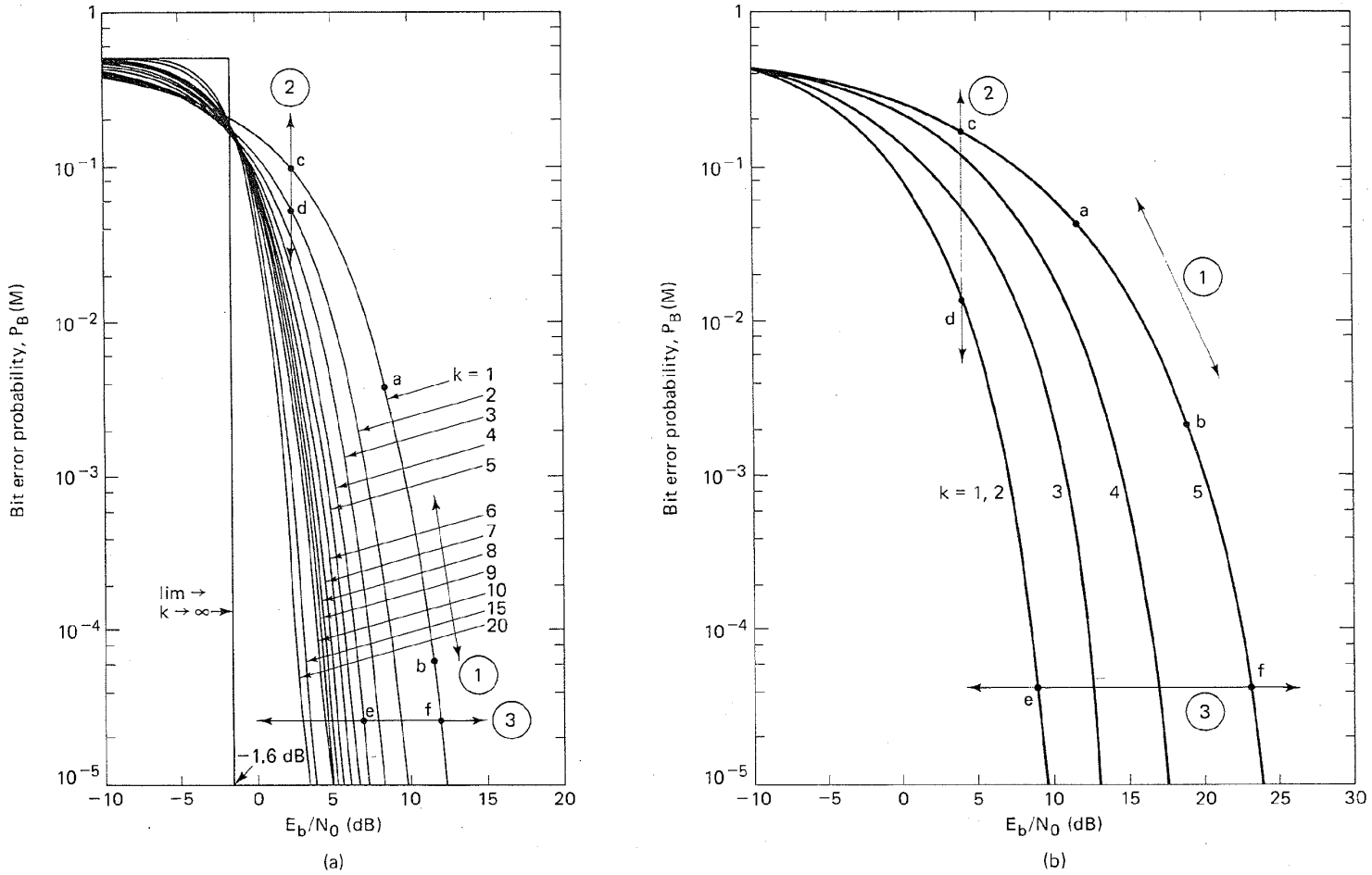


Figure 7.1 Bit error probability versus E_b/N_0 for coherently detected M-ary signaling. (a) Orthogonal signaling. (b) Multiple phase signaling.

7.3 NYQUIST MINIMUM BANDWIDTH

Every realizable system having some nonideal filtering will suffer from intersymbol interference (ISI)—the tail of one pulse spilling over into adjacent symbol intervals so as to interfere with correct detection. Nyquist [1] showed that, in theory, R_s symbols per second could be detected without ISI in an $R_s/2$ hertz minimum bandwidth (Nyquist bandwidth); this is a basic theoretical constraint, limiting the designer's goal to expend as little bandwidth as possible (see Section 2.11). In practice, R_s hertz is typically required for the transmission of R_s symbols per second. In other words, *typical* digital communication throughput, without ISI, is limited to 1 symbol/s per hertz. The modulation or coding system assigns to each symbol, of its set of M symbols, a k-bit meaning, where $M = 2^k$. For a signaling scheme with a fixed bandwidth, such as MPSK, as k increases, the allowable data rate, R, increases, and hence the bandwidth efficiency, R/W, measured in bits per second per hertz, also increases. For example, movement along line 3, from point e to point f in Figure 7.1b represents trading E_b/N_0 for a reduced bandwidth requirement. In other words, with the same system bandwidth one can transmit at an increased data rate, hence at an increased R/W.

7.4 SHANNON–HARTLEY CAPACITY THEOREM

Shannon [2] showed that the system capacity, C, of a channel perturbed by additive white Gaussian noise (AWGN) is a function of the average received signal power, S, the average noise power, N, and the bandwidth, W. The capacity relationship (Shannon–Hartley theorem) can be stated as

$$C = W \log_2 \left(1 + \frac{S}{N}\right) \tag{7.1}$$

When W is in hertz and the logarithm is taken to the base 2, as shown, the capacity is given in bits/s. It is theoretically possible to transmit information over such a channel at any rate, R, where $R \leq C$, with an *arbitrarily small* error probability by using a sufficiently complicated coding scheme. For an information rate $R > C$, it is not possible to find a code that can achieve an arbitrarily small error probability. Shannon's work showed that the values of S, N, and W *set a limit on transmission rate, not on error probability*. Shannon [3] used Equation (7.1) to graphically exhibit a bound for the achievable performance of practical systems. This plot, shown in Figure 7.2, gives the normalized channel capacity C/W in bits/s/Hz as a function of the channel signal-to-noise ratio (SNR). A related plot, shown in Figure 7.3, indicates the normalized channel bandwidth W/C in Hz/bits/s as a function of SNR in the channel. Figure 7.3 is sometimes used to illustrate the power–bandwidth trade-off inherent in the ideal channel. However, it is not a pure trade-off [4] because the detected noise power is proportional to bandwidth.

$$N = N_0 W \tag{7.2}$$

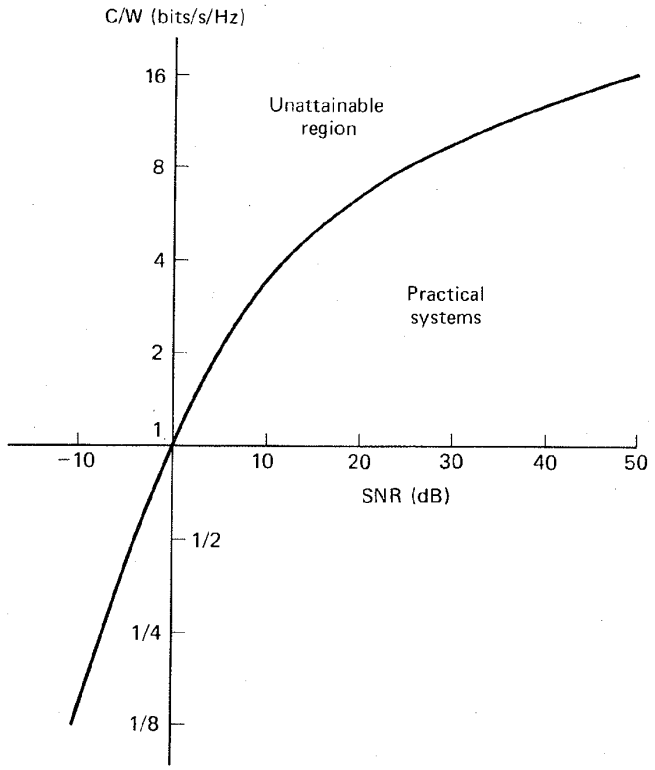


Figure 7.2 Normalized channel capacity versus channel SNR.

Substituting Equation (7.2) into Equation (7.1) and rearranging terms yields

$$\frac{C}{W} = \log_2\left(1 + \frac{S}{N_0 W}\right) \tag{7.3}$$

For the case where transmission bit rate is equal to channel capacity, $R = C$, we can use the identity presented in Equation (3.94) to write

$$\frac{S}{N_0 C} = \frac{E_b}{N_0} \tag{7.4}$$

Hence we can modify Equation (7.3) as follows:

$$\frac{C}{W} = \log_2\left[1 + \frac{E_b}{N_0}\left(\frac{C}{W}\right)\right] \tag{7.5}$$

$$2^{C/W} = 1 + \frac{E_b}{N_0}\left(\frac{C}{W}\right)$$

$$\frac{E_b}{N_0} = \frac{W}{C}(2^{C/W} - 1) \tag{7.6}$$

Figure 7.4 is a plot of W/C versus E_b/N_0 in accordance with Equation (7.6).

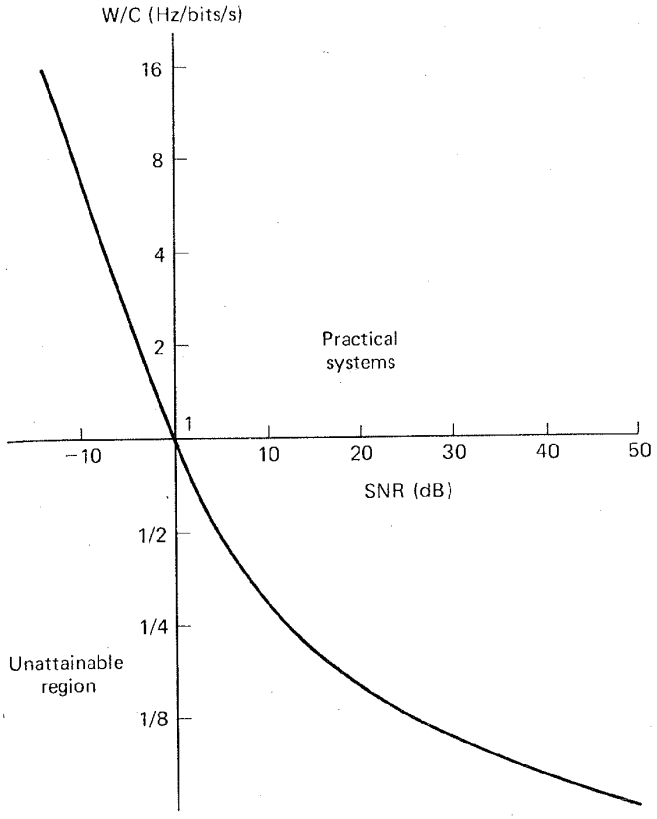


Figure 7.3 Normalized channel bandwidth versus channel SNR.

The asymptotic behavior of this curve as $C/W \to 0$ (or $W/C \to \infty$) is discussed in the next section.

7.4.1 Shannon Limit

There exists a limiting value of E_b/N_0 below which there can be no error-free communication at any information rate. Using the identity

$$\lim_{x \to 0} (1 + x)^{1/x} = e$$

we can calculate the limiting value of E_b/N_0 as follows. Let

$$x = \frac{E_b}{N_0}\left(\frac{C}{W}\right)$$

Then from Equation (7.5),

$$\frac{C}{W} = x \log_2 (1 + x)^{1/x}$$

$$1 = \frac{E_b}{N_0} \log_2 (1 + x)^{1/x}$$

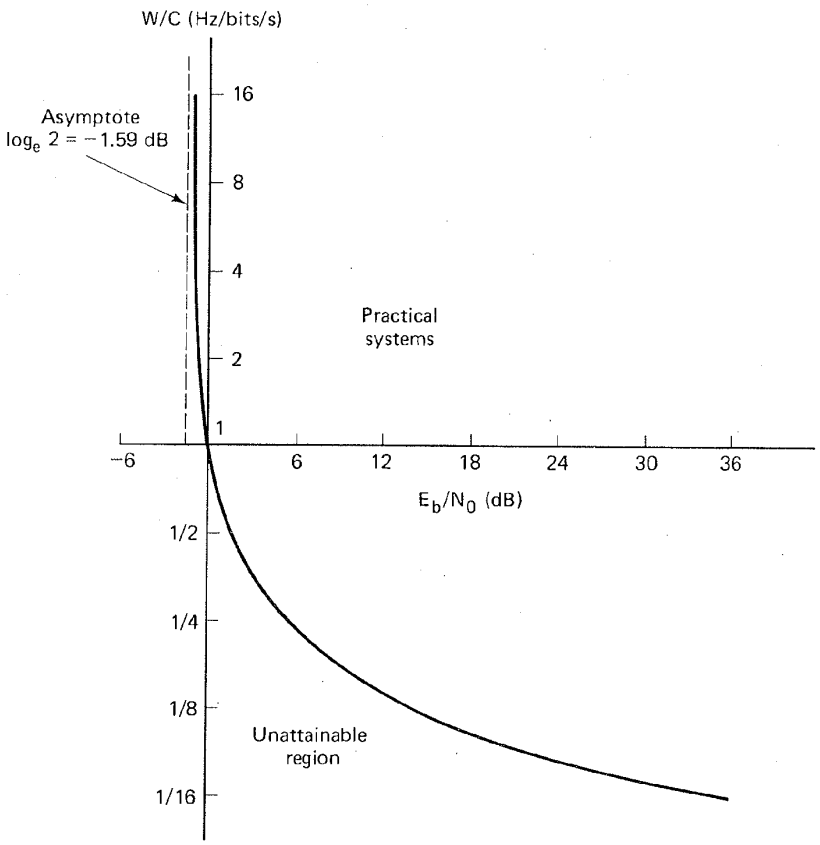


Figure 7.4 Normalized channel bandwidth versus channel E_b/N_0.

In the limit, as $C/W \rightarrow 0$, we get

$$\frac{E_b}{N_0} = \frac{1}{\log_2 e} = 0.693$$
$$\text{or, in decibels} = -1.59 \text{ dB} \tag{7.7}$$

This value of E_b/N_0 is called the *Shannon limit*. On Figure 7.1a the Shannon limit is the P_B versus E_b/N_0 curve corresponding to $k \rightarrow \infty$. The curve is discontinuous, going from a value of $P_B = \frac{1}{2}$ to $P_B = 0$ at $E_b/N_0 = -1.59$ dB. It is not possible in practice to reach the Shannon limit, because as k increases without bound, the bandwidth requirement and the implementation complexity increase without bound. Shannon's work provided a theoretical proof for the existence of codes that could improve the P_B performance, or reduce the E_b/N_0 required, from the levels of the uncoded binary modulation schemes to levels approaching the limiting curve. For a bit error probability of 10^{-5}, binary phase shift keying (BPSK) modulation requires an E_b/N_0 of 9.6 dB (the optimum uncoded binary modulation). Therefore, Shannon's work promised the existence of a theoretical performance

improvement of 11.2 dB over the performance of optimum uncoded binary modulation, through the use of coding techniques. Today, most of that promised improvement (approximately 7 dB) is realizable [5]. Optimum system design can best be described as a search for rational compromises or trade-offs among the various constraints and conflicting goals. The modulation and coding trade-off, that is, the selection of modulation and coding techniques to make the best use of transmitter power and channel bandwidth, is important, since there are strong incentives to reduce the cost of generating power and to conserve the radio spectrum.

7.4.2 Entropy

To design a communications system with a specified message handling capability, we need a metric for measuring the information content to be transmitted. Shannon [2] developed such a metric, H, called the entropy of the message source (having n possible outputs). *Entropy* is defined as the average amount of information per source output and is expressed by

$$H = -\sum_{i=1}^{n} p_i \log_2 p_i \qquad \text{bits/source output} \tag{7.8}$$

where p_i is the probability of the ith output and $\sum p_i = 1$. In the case of a binary message or a source having only two possible outputs, with probabilities p and $q = (1 - p)$, the entropy is written

$$H = -(p \log_2 p + q \log_2 q) \tag{7.9}$$

and is plotted versus p in Figure 7.5.

The quantity H has a number of interesting properties, including the following:

1. When the logarithm in Equation (7.8) is taken to the base 2, as shown, the unit for H is average bits per event. The unit *bit*, here, is a measure of *information content* and is not to be confused with the term "bit," meaning "binary digit."
2. The term "entropy" has the same uncertainty connotation as it does in certain formulations of statistical mechanics. For the information source with two equally likely possibilities (e.g., the flipping of a fair coin), it can be seen from Figure 7.5 that the uncertainty in the event, and hence the average information content, is maximum. As the probabilities depart from the equally likely case, the average information content decreases. In the limit, when one of the probabilities goes to zero, H also goes to zero. We know the result before the event happens, so the result conveys no additional information.
3. To illustrate that information content is related to a priori probability (if the a priori message probability at the receiver is zero or one, we need not send the message), consider the following example: At the end of her nine-month

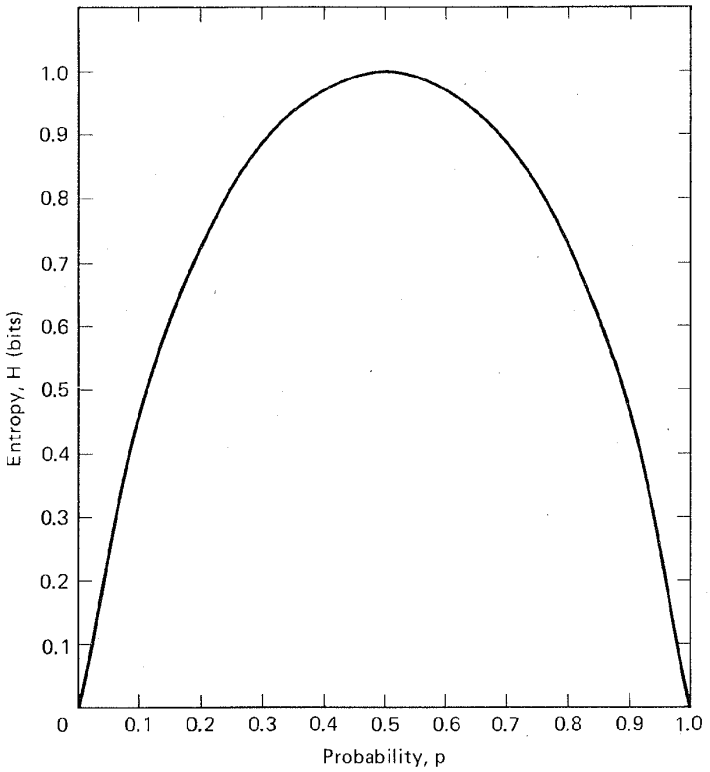


Figure 7.5 Entropy versus probability (two events).

pregnancy, a woman enters the delivery room of a local hospital to give birth. Her husband waits anxiously in the waiting room. After some time, a physician approaches the husband and says: "Congratulations, you are the father of a child." How much information has the physician given the father *beyond the medical outcome*? Almost none; the father has known with virtual certainty that a child was forthcoming. Had the physician said, "you are the father of a boy" or "you are the father of a girl," he would have transmitted 1 bit of information, since there was a 50% chance that the child could have been a boy or a girl.

Example 7.1 Average Information Content in the English Language

(a) Calculate the average information in bits/character for the English language, assuming that each of the 26 characters in the alphabet occurs with equal likelihood. Neglect spaces and punctuation.

(b) Since the alphabetic characters do not appear with equal frequency in the English language (or any other language), the answer to part (a) will represent an upper bound on average information content per character. Repeat part (a) under the assumption that the alphabetic characters occur with the following probabilities:

$p = 0.10$: for the letters a, e, o, t

$p = 0.07$: for the letters h, i, n, r, s

$p = 0.02$: for the letters c, d, f, l, m, p, u, y

$p = 0.01$: for the letters b, g, j, k, q, v, w, x, z

Solution

(a) $$H = -\sum_{i=1}^{26} \frac{1}{26} \log_2 \left(\frac{1}{26}\right)$$
$$= 4.7 \text{ bits/character}$$

(b) $$H = -(4 \times 0.1 \log_2 0.1 + 5 \times 0.07 \log_2 0.07 + 8 \times 0.02 \log_2 0.02 + 9 \times 0.01 \log_2 0.01)$$
$$= 4.17 \text{ bits/character}$$

If we want to express the 26 letters of the alphabet with some binary-digit coding scheme, we generally need five binary digits for each character. Example 7.1 demonstrates that there may be a way to encode the English language with a fewer number of binary digits per character, *on the average,* by exploiting the fact that the average amount of information contained within each character is less than 5 bits. The subject of source coding, which deals with this exploitation, is treated in Chapter 11.

7.4.3 Equivocation and Effective Transmission Rate

Suppose that we are transmitting information at a rate of 1000 binary symbols/s over a binary symmetric channel (defined in Section 5.3.1), and that the a priori probability of transmitting either a one or a zero is equally likely. Suppose also that the noise in the channel is so great that the probability of receiving a one is $\frac{1}{2}$, whatever was transmitted, and similarly for receiving a zero. In such a case, half the received symbols would be correct *due to chance alone,* and the system might appear to be providing 500 bits/s while actually no information is being received at all. Equally "good" reception could be obtained by dispensing with the channel entirely and "flipping a coin" within the receiver. The proper correction to apply to the amount of information transmitted is the amount of information that is lost in the channel. Shannon [2] uses a correction factor called *equivocation* to account for the uncertainty in the received signal. Equivocation is defined as the *conditional entropy* of the message X, given Y, as shown below:

$$\begin{aligned} H(X|Y) &= -\sum_{X,Y} P(X, Y) \log_2 P(X|Y) \\ &= -\sum_{Y} P(Y) \sum_{X} P(X|Y) \log_2 P(X|Y) \end{aligned} \tag{7.10}$$

where X is the transmitted source message, Y is the received signal, $P(X, Y)$ is the joint probability of X and Y, and $P(X|Y)$ is the conditional probability of X given Y. Equivocation can be thought of as the uncertainty that message X was

sent, having received Y. For an *error-free channel*, $H(X|Y) = 0$, because having received Y, there is complete certainty about the message X. However, for a channel with a nonzero probability of symbol error, $H(X|Y) > 0$, because the channel introduces uncertainty. Consider a binary sequence, X, where the a priori source probabilities are $P(X = 1) = P(X = 0) = \frac{1}{2}$, and where, on the average, the channel produces one error in a received sequence of 100 bits ($P_B = 0.01$). Using Equation (7.10), the equivocation $H(X|Y)$ is expressed as

$$\begin{aligned} H(X|Y) &= -[(1 - P_B) \log_2 (1 - P_B) + P_B \log_2 P_B] \\ &= -(0.99 \log_2 0.99 + 0.01 \log_2 0.01) \\ &= 0.081 \text{ bit/received symbol} \end{aligned}$$

Thus, the channel introduces 0.081 bit of uncertainty to each received symbol.

Shannon showed that the average effective information content, H_{eff}, at the receiver, is obtained by subtracting the equivocation from the entropy of the source. Therefore,

$$H_{\text{eff}} = H(X) - H(X|Y) \tag{7.11}$$

For a system transmitting equally likely binary symbols, the entropy, $H(X)$, is 1 bit/symbol. When the symbols are received with $P_B = 0.01$ the equivocation is 0.081 bit/received symbol as was calculated above. Then using Equation (7.11), the effective entropy of the received signal, H_{eff}, is

$$H_{\text{eff}} = 1 - 0.081 = 0.919 \text{ bit/received symbol}$$

Thus, if $R = 1000$ binary symbols transmitted per second, for example, the effective information bit rate, R_{eff}, can be expressed as

$$\begin{aligned} R_{\text{eff}} &= RH_{\text{eff}} \\ &= 1000 \text{ symbols/s} \times 0.919 \text{ bit/symbol} = 919 \text{ bits/s} \end{aligned} \tag{7.12}$$

Notice that in the extreme case where $P_B = 0.5$,

$$\begin{aligned} H(X|Y) &= -(0.5 \log_2 0.5 + 0.5 \log_2 0.5) \\ &= 1 \text{ bit/symbol} \end{aligned}$$

and, applying Equations (7.12) and (7.11) to the $R = 1000$ symbols/s example, yields

$$R_{\text{eff}} = 1000 \text{ symbols/s } (1 - 1) = 0 \text{ bit/s}$$

as should be expected.

Example 7.2 Apparent Contradiction in the Shannon Limit

Plots of P_B versus E_b/N_0 typically display a smooth increase of P_B as E_b/N_0 is decreased. For example, the bit error probability for the curves in Figure 7.1 shows P_B *tending* to 0.5 in the limit as E_b/N_0 approaches zero. Thus there is apparently always a nonvanishing information rate, regardless of how small E_b/N_0 becomes. This *appears to contradict* the Shannon limit of $E_b/N_0 = -1.59$ dB, below which

no error-free information rate can be supported per unit bandwidth, or below which even an infinite bandwidth cannot support a finite information rate (see Figure 7.4).

(a) Suggest a way of resolving the apparent contradiction.
(b) Show how Shannon's equivocation correction can resolve it for a binary PSK system where the source has an entropy of 1 bit/symbol. Consider that the operating point on Figure 7.1b corresponds to $E_b/N_0 = 0.1$ (−10 dB).

Solution

(a) The value of E_b, traditionally used in link calculations for practical systems, is invariably the received signal energy per *transmitted symbol*. However, the meaning of E_b in Equation (7.6) is the signal energy per bit of *received information*. The information loss caused by the noisy channel must be taken into account to resolve the apparent contradiction.
(b) Following Equation (3.84) for BPSK,

$$P_B = Q(\sqrt{2E_b/N_0}) = Q(0.447)$$

where Q is defined in Equation (2.42) and tabulated in Table B.1. From the tabulation, P_B is found to be 0.33. Next, we solve for the equivocation and effective entropy:

$$\begin{aligned} H(X|Y) &= -[(1 - P_B)\log_2(1 - P_B) + P_B \log_2 P_B] \\ &= -(0.67 \log_2 0.67 + 0.33 \log_2 0.33) \\ &= 0.915 \text{ bit/symbol} \end{aligned}$$

$$\begin{aligned} H_{\text{eff}} &= H(X) - H(X|Y) \\ &= 1 - 0.915 \\ &= 0.085 \text{ bit/symbol} \end{aligned}$$

Hence

$$\left(\frac{E_b}{N_0}\right)_{\text{eff}} = \frac{(E_b/N_0) \quad \text{joules per symbol/watts per hertz}}{H_{\text{eff}} \quad \text{bits/symbol}}$$

$$= \frac{0.1}{0.085} = 1.176 \frac{\text{joules per bit}}{\text{watts/Hz}}$$

$$= 0.7 \text{ dB}$$

Thus the effective value of E_b/N_0 is equal to 0.7 dB per received information bit, which is well above Shannon's limit of −1.59 dB.

7.5 BANDWIDTH-EFFICIENCY PLANE

Using Equation (7.6), we can plot normalized channel bandwidth W/C in Hz/bits/s versus E_b/N_0, as shown in Figure 7.4. Here, with the abscissa taken as E_b/N_0, we see the *true power–bandwidth trade-off* at work. It can be shown [4] that well-designed systems tend to operate near the "knee" of this power–bandwidth trade-off curve for the ideal ($R = C$) channel. Actual systems are frequently within 10 dB or less of the performance of the ideal. The existence of the knee means that

systems seeking to reduce the channel bandwidth they occupy or to reduce the signal power they require must make an increasingly unfavorable exchange in the other parameter. For example, from Figure 7.4, an ideal system operating at an E_b/N_0 of 1.8 dB and using a normalized bandwidth of 0.5 Hz/bits/s would have to increase E_b/N_0 to 20 dB to reduce the bandwidth occupancy to 0.1 Hz/bits/s. Trade-offs in the other direction are similarly inequitable.

Using Equation (7.6), we can also plot C/W versus E_b/N_0. This relationship is shown plotted on the R/W versus E_b/N_0 plane in Figure 7.6. We shall denote

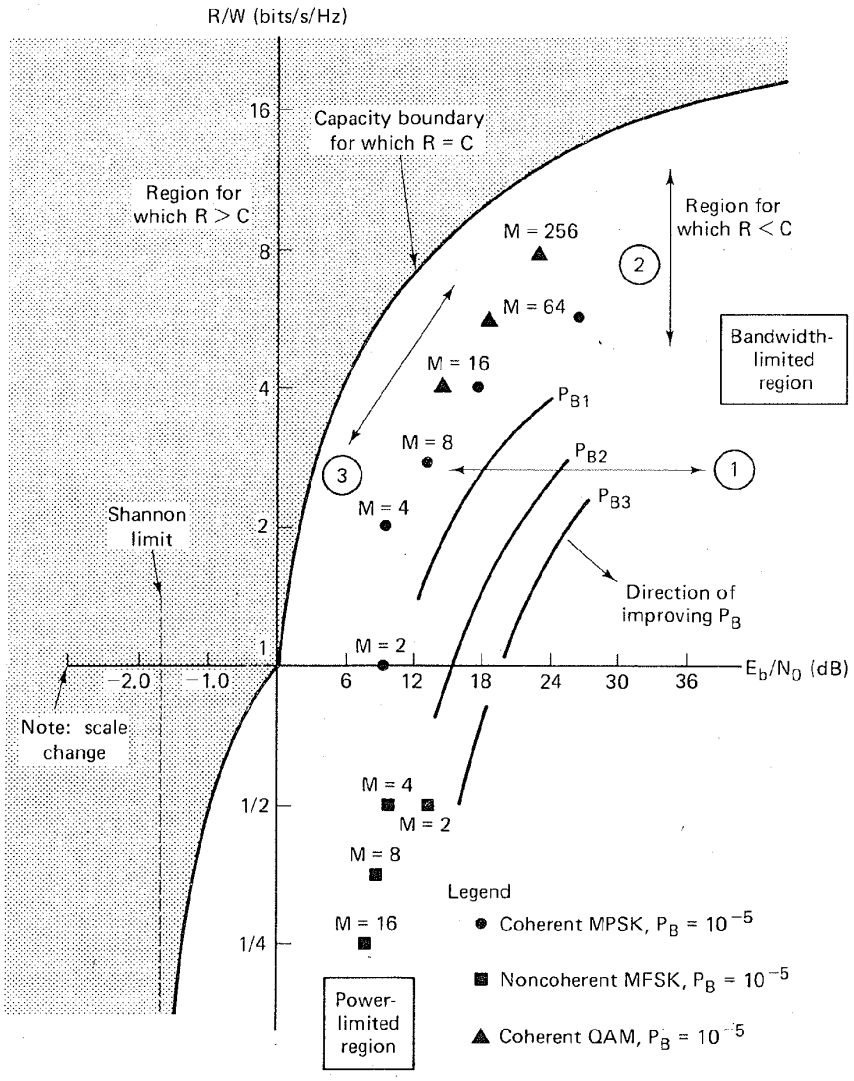


Figure 7.6 Bandwidth-efficiency plane.

this plane as the *bandwidth-efficiency plane*. The ordinate, R/W, is a measure of how much data can be communicated in a specified bandwidth within a given time; it therefore reflects how efficiently the bandwidth resource is utilized. The abscissa is E_b/N_0 in units of decibels. For the case where $R = C$ in Figure 7.6, the curve represents a boundary that separates a region characterizing practical communication systems from a region where such communication systems are not theoretically possible. Like Figure 7.2, the bandwidth-efficiency plane in Figure 7.6 sets the limiting performance that can be achieved by practical systems. Since the abscissa in Figure 7.6 is E_b/N_0 rather than SNR, Figure 7.6 is more useful for comparing digital communication modulation and coding trade-offs than is Figure 7.2.

7.5.1 Bandwidth Efficiency of MPSK and MFSK Modulation

On the bandwidth-efficiency plane of Figure 7.6 are plotted the operating points for coherent MPSK modulation at a bit error probability of 10^{-5}. We assume Nyquist (ideal rectangular) filtering at baseband, so that the minimum double-sideband (DSB) bandwidth at an intermediate frequency (IF) is $W_{IF} = 1/T$, where T is the symbol duration. Thus the bandwidth efficiency is $R/W = \log_2 M$, where M is the symbol set size. For realistic channels and waveforms, the performance must be reduced to account for the bandwidth increase required to implement realizable filters. Notice that for MPSK modulation, R/W increases with increasing M. Notice also that the location of the MPSK points indicates that BPSK ($M = 2$) and quaternary PSK or QPSK ($M = 4$) require the same E_b/N_0. That is, for the same value of E_b/N_0, QPSK has a bandwidth efficiency of 2 bits/s/Hz, compared to 1 bit/s/Hz for BPSK. This unique feature stems from the fact that QPSK is effectively a composite of two BPSK signals transmitted on orthogonal components of the carrier.

Also plotted on the bandwidth-efficiency plane of Figure 7.6 are the operating points for noncoherent MFSK modulation at a bit error probability of 10^{-5}. We assume that the IF transmission bandwidth is $W_{IF} = M/T$, and thus the bandwidth efficiency is $R/W = k/M$. Notice that for MFSK modulation, R/W decreases with increasing M. Notice also that the position of the MFSK points indicates that BFSK ($M = 2$) and quaternary FSK ($M = 4$) have the same bandwidth efficiency, even though the former requires greater E_b/N_0 for the same error probability. The bandwidth efficiency varies with the modulation index (tone spacing in hertz divided by bit rate). Under the assumption that an equal increment of bandwidth is required for each MFSK tone the system uses, it can be seen that for $M = 2$, the bandwidth efficiency is 1 bit/s/2 Hz or $\frac{1}{2}$, and for $M = 4$, similarly, the R/W is 2 bits/s/4 Hz or $\frac{1}{2}$.

Operating points for coherent quadrature amplitude modulation (QAM) are also plotted in Figure 7.6. Of the modulations shown, QAM is clearly the most bandwidth efficient; it is treated in greater detail in Section 7.9.3.

7.5.2 Analogies between Bandwidth-Efficiency and Error Probability Planes

The bandwidth-efficiency plane in Figure 7.6 is analogous to the error probability plane in Figure 7.1. The Shannon limit of the Figure 7.1 plane is analogous to the capacity boundary of the Figure 7.6 plane. The curves in Figure 7.1 were referred to as equibandwidth curves. In Figure 7.6, we can analogously describe equi-error-probability curves for various modulation and coding schemes. The curves, labeled P_{B1}, P_{B2}, and P_{B3}, are hypothetical constructions for some arbitrary modulation and coding scheme; the P_{B1} curve represents the largest error probability of the three curves, and the P_{B3} curve represents the smallest. The general direction in which the curves move for improved P_B is indicated on the figure.

Just as potential trade-offs among P_B, E_b/N_0, and W were considered for the error probability plane, the same trade-offs can be considered on the bandwidth efficiency plane. The potential trade-offs are seen in Figure 7.6 as changes in operating point in the direction shown by the arrows. Movement of the operating point along line 1 can be viewed as trading P_B for E_b/N_0, with R/W fixed. Similarly, movement along line 2 is seen as trading P_B for W (or R/W), with E_b/N_0 fixed. Finally, movement along line 3 illustrates trading W (or R/W) for E_b/N_0, with P_B fixed. In Figure 7.6, as in Figure 7.1, movement along line 1 can be effected by increasing or decreasing the available E_b/N_0. However, movement along line 2 or line 3 requires changes in the system modulation or coding scheme.

The two primary communications resources are the transmitted power and the channel bandwidth. In many communication systems, one of these resources may be more precious than the other, and hence most systems can be classified as either power limited or bandwidth limited. In *power-limited systems,* coding schemes can be used to save power at the expense of bandwidth, whereas in *bandwidth-limited systems,* spectrally efficient modulation techniques can be used to save bandwidth at the expense of power.

7.6 POWER-LIMITED SYSTEMS

For the case of power-limited systems, systems in which power is scarce but system bandwidth is available (e.g., a space communication link), the following trade-offs might be made: (1) improved P_B can be achieved by expending bandwidth (for a given E_b/N_0); or (2) required E_b/N_0 can be reduced by expending bandwidth (for a given P_B). The error probability plane of Figure 7.1a can be very useful for examining these potential trade-offs. It is on such a plane that we can verify whether or not a candidate modulation or code offers improvement in required E_b/N_0 for a particular channel and for a specified P_B (or whether the modulation or code offers improvement in P_B for a given E_b/N_0).

7.7 BANDWIDTH-LIMITED SYSTEMS

Any digital scheme that transmits $\log_2 M$ bits in T seconds using a bandwidth of W hertz operates at a bandwidth efficiency of $R/W = (\log_2 M)/WT$ bits/s/Hz. From this expression it can be seen that the smaller the WT product, the more bandwidth efficient will be the system. Signals with small WT products are more often used with bandwidth-limited systems—systems in which channel bandwidth is constrained but power is available. For this case the usual objective is to design the link so as to maximize the transmitted information rate over the bandlimited channel, at the expense of E_b/N_0 (while maintaining a specified value of P_B). For bandlimited operation, bandwidth efficiency is a useful criterion of system performance, and the bandwidth-efficiency plane of Figure 7.6 is useful for examining potential trade-offs.

Two regions, the bandwidth-limited region and the power-limited region, are shown on the bandwidth efficiency plane of Figure 7.6. Notice that the desirable trade-offs associated with each of these regions are not equitable. For the bandwidth-limited region, large R/W is desired; however, as E_b/N_0 is increased, the capacity boundary curve flattens out and ever-increasing amounts of additional E_b/N_0 are required to achieve improvement in R/W. A similar relationship is at work in the power-limited region. Here a savings in E_b/N_0 is desired, but the capacity boundary curve is steep; to achieve a small reduction in required E_b/N_0 requires a large reduction in R/W.

7.8 MODULATION AND CODING TRADE-OFFS

Figure 7.7 is useful in pointing out analogies between the two performance planes, the error probability plane of Figure 7.1 and the bandwidth efficiency plane of Figure 7.6. Figure 7.7a and b represent the same planes as Figures 7.1 and 7.6, respectively. They have been redrawn as symmetrical, by choosing appropriate scales. In each case the arrows and their labels describe the general effect of moving an operating point in the direction of the arrow by means of appropriate modulation and coding techniques. The notations G, C, and F stand for the trade-off considerations "*G*ained or achieved," "*C*ost or expended," and "*F*ixed or unchanged," respectively. The parameters being traded are P_B, W, R/W, and P (power or S/N). Just as the movement of an operating point toward the Shannon limit in Figure 7.7a can achieve improved P_B or reduced required transmitter power at the cost of bandwidth, so too movement toward the capacity boundary in Figure 7.7b can improve bandwidth efficiency at the cost of increased required power or degraded P_B.

Most often, these trade-offs are examined with a fixed P_B (constrained by the system requirement) in mind. Therefore, the most interesting arrows are those having bit error probability (marked F: P_B). There are four such arrows on Figure

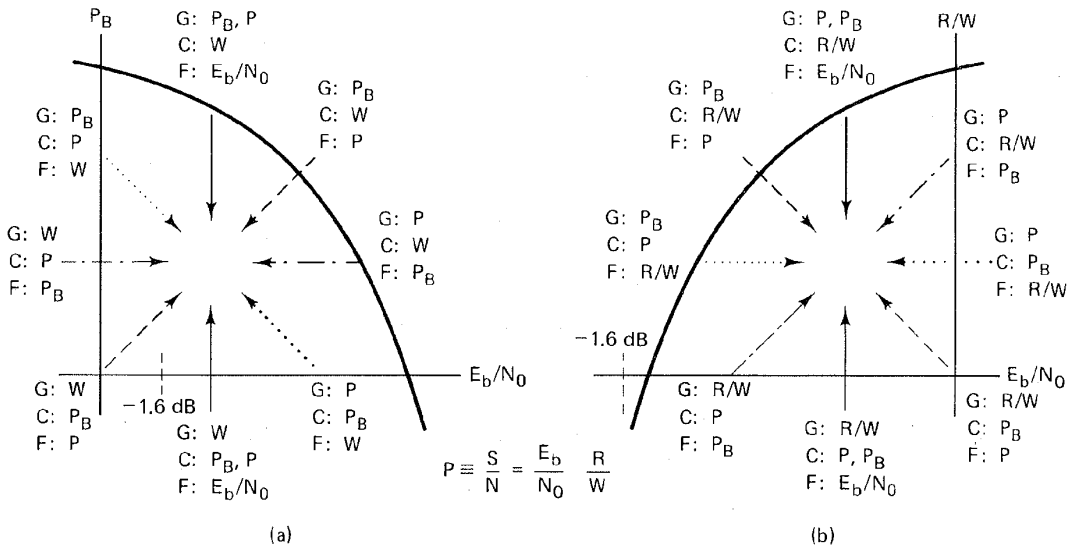


Figure 7.7 Modulation/coding trade-offs. (a) Error probability plane. (b) Bandwidth-efficiency plane.

7.7, two on the error probability plane and two on the bandwidth-efficiency plane. Arrows marked with the same pattern indicate correspondence between the two planes. System operation can be characterized by either of these two planes. The planes represent two ways of looking at some of the key system parameters; each plane highlights slightly different aspects of the overall design problem. The error probability plane tends to be most useful with *power-limited systems,* where as we move from curve to curve, the bandwidth requirements are only inferred, while the bit error probability is clearly displayed. The bandwidth efficiency plane is generally more useful for examining *bandwidth-limited systems*; here as we move from curve to curve, the bit error probability is only inferred, but the bandwidth requirements are explicit.

The two system trade-off planes, error probability and bandwidth efficiency, have been presented *heuristically* with simple examples (orthogonal and multiple phase signaling) to provide some insight into the design issues of trading-off error probability, bandwidth, and power. The ideas are useful for *most modulation and coding schemes,* with the following caveat. For *some* codes or combined modulation and coding schemes, the performance curves *do not move as predictably* as those for the examples chosen here. The reason has to do with the strength and bandwidth expansion features of the particular code. For example, the performance of coherent PSK combined with several codes was illustrated in Figure 5.23. Examine the curves characterizing the two BCH codes, (127, 64) and (127, 36). It should be clear from their relative positions that the (127, 64) code manifests *greater coding gain* than the (127, 36) code. This violates our expectations since, within the same block size, the latter code has greater redundancy

(requires more bandwidth expansion) than the former. Also, in the area of trellis-coded modulation covered in Section 7.10.6, we consider codes that provide coding gain without any bandwidth expansion. Performance curves for such coding schemes will also behave differently from the curves of most modulation and coding schemes discussed so far.

7.9 BANDWIDTH-EFFICIENT MODULATION

The primary objective of spectrally efficient modulation techniques is to maximize bandwidth efficiency. The increasing demand for digital transmission channels has led to the investigation of spectrally efficient modulation techniques [6] to maximize bandwidth efficiency and thus help ameliorate the spectral congestion problem.

Some systems have additional modulation requirements besides spectral efficiency. For example, satellite systems with highly nonlinear transponders require a constant envelope modulation. This is because the nonlinear transponder produces extraneous sidebands when passing a signal with amplitude fluctuations (due to a mechanism called AM-to-PM conversion). These sidebands deprive the information signals of some of their portion of transponder power, and also can interfere with nearby channels (adjacent channel interference) or with other communication systems (co-channel interference). *Offset QPSK* (OQPSK) and *Minimum shift keying* (MSK) are two examples of constant envelope modulation schemes that are attractive for systems using nonlinear transponders.

7.9.1 QPSK and Offset QPSK Signaling

Figure 7.8 illustrates the partitioning of a typical pulse stream for QPSK modulation. Figure 7.8a shows the original data stream $d_k(t) = d_0, d_1, d_2, \ldots$ consisting of bipolar pulses; that is, the values of $d_k(t)$ are $+1$ or -1, representing binary one and zero, respectively. This pulse stream is divided into an in-phase stream, $d_I(t)$, and a quadrature stream, $d_Q(t)$, illustrated in Figure 7.8b, as follows:

$$\begin{aligned} d_I(t) &= d_0, d_2, d_4, \ldots \quad \text{(even bits)} \\ d_Q(t) &= d_1, d_3, d_5, \ldots \quad \text{(odd bits)} \end{aligned} \tag{7.13}$$

Note that $d_I(t)$ and $d_Q(t)$ each have half the bit rate of $d_k(t)$. A convenient orthogonal realization of a QPSK waveform, $s(t)$, is achieved by amplitude modulating the in-phase and quadrature data streams onto the cosine and sine functions of a carrier wave, as follows:

$$s(t) = \frac{1}{\sqrt{2}} d_I(t) \cos\left(2\pi f_0 t + \frac{\pi}{4}\right) + \frac{1}{\sqrt{2}} d_Q(t) \sin\left(2\pi f_0 t + \frac{\pi}{4}\right) \tag{7.14}$$

Using the trigonometric identities shown in Equations (D.5) and (D.6), Equation

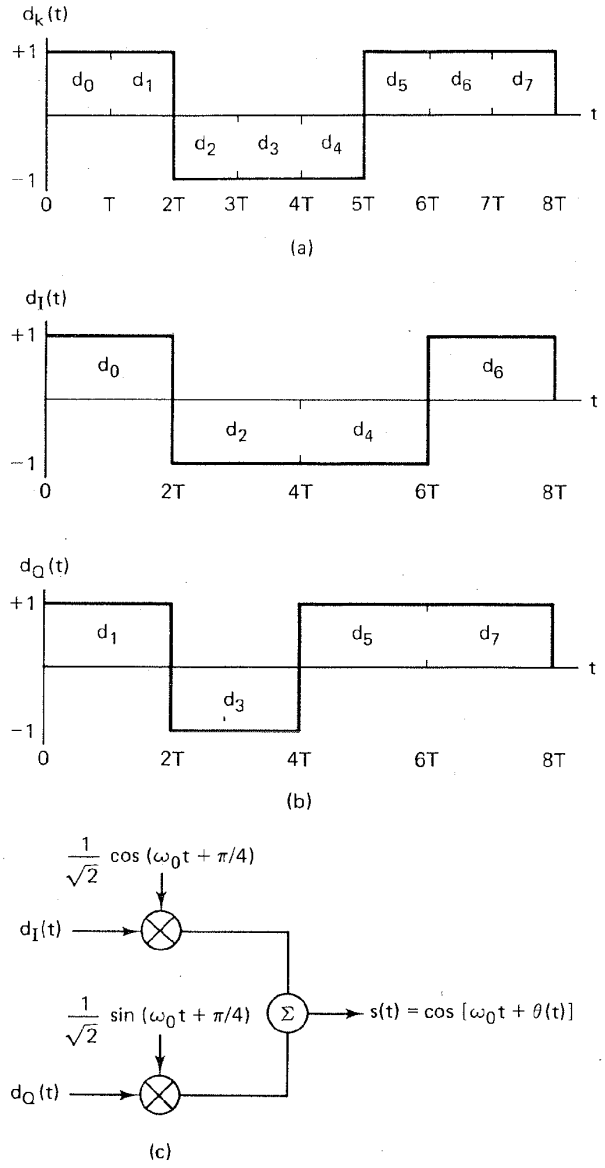


Figure 7.8 QPSK modulation.

(7.14) can also be written as

$$s(t) = \cos [2\pi f_0 t + \theta(t)] \tag{7.15}$$

The QPSK modulator is shown in the block diagram of Figure 7.8c. The pulse stream $d_I(t)$ amplitude-modulates the cosine function with an amplitude of $+1$ or -1. This is equivalent to shifting the phase of the cosine function by 0 or π; consequently, this produces a BPSK waveform. Similarly, the pulse stream $d_Q(t)$ modulates the sine function, yielding a BPSK waveform orthogonal to the cosine

function. The summation of these two orthogonal components of the carrier yields the QPSK waveform. The value of $\theta(t)$ will correspond to one of the four possible combinations of $d_I(t)$ and $d_Q(t)$ in Equation (7.14). These values are: $\theta(t) = 0°$, $\pm 90°$, or $180°$, and the resulting signal vectors are seen in the signal space illustrated in Figure 7.9. Because $\cos(2\pi f_0 t + \pi/4)$ and $\sin(2\pi f_0 t + \pi/4)$ are orthogonal, the two BPSK signals can be detected separately.

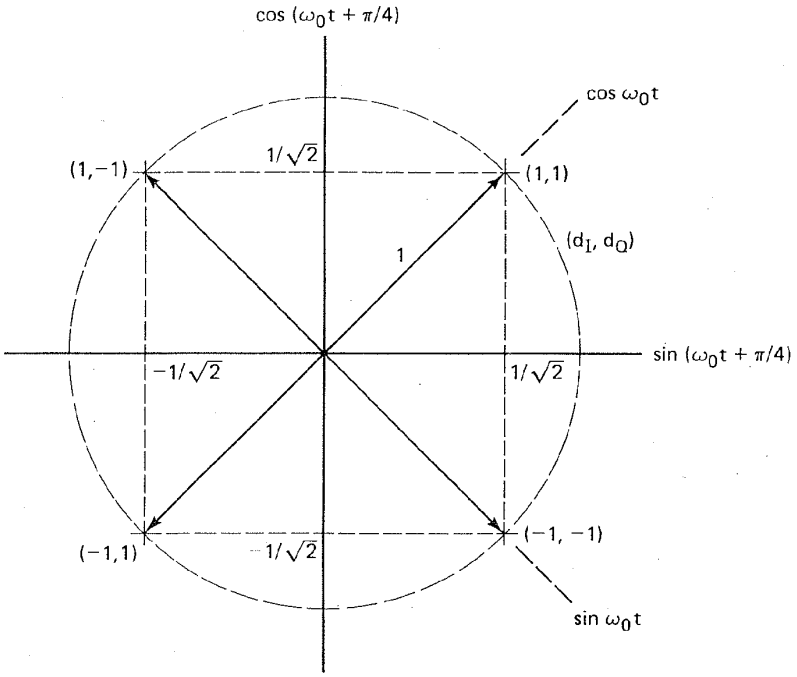


Figure 7.9 Signal space for QPSK and OQPSK.

Offset QPSK (OQPSK) signaling can also be represented by Equations (7.14) and (7.15); the difference between the two modulation schemes, QPSK and OQPSK, is only in the *alignment* of the two baseband waveforms. As shown in Figure 7.8, the duration of each original pulse is T (Figure 7.8a), and hence in the partitioned streams of Figure 7.8b, the duration of each pulse is $2T$. In standard QPSK, the odd and even pulse streams are both transmitted at the rate of 1/2T bits/s and are synchronously aligned, such that their transitions coincide, as shown in Figure 7.8b. In OQPSK, sometimes called *staggered QPSK* (SQPSK), there is the same data stream partitioning and orthogonal transmission; the difference is that the timing of the pulse stream $d_I(t)$ and $d_Q(t)$ is shifted such that the alignment of the two streams is offset by T. Figure 7.10 illustrates this offset.

In standard QPSK, due to the coincident alignment of $d_I(t)$ and $d_Q(t)$, the carrier phase can change only once every $2T$. The carrier phase during any $2T$ interval can be any one of the four phases shown in Figure 7.9, depending on the values of $d_I(t)$ and $d_Q(t)$ during that interval. During the next $2T$ interval, if neither pulse stream changes sign, the carrier phase remains the same. If only one of the

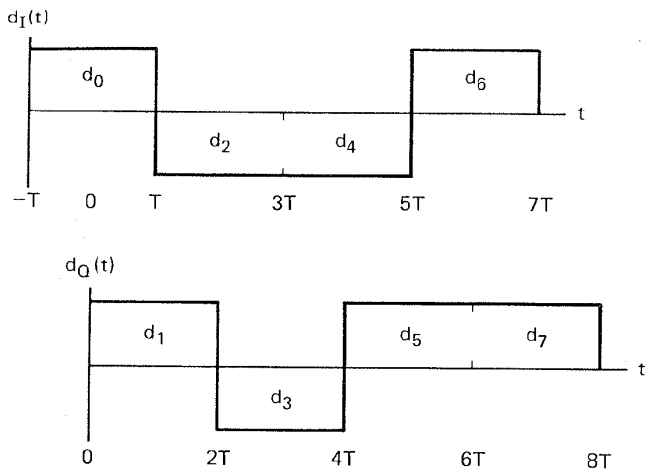


Figure 7.10 Offset QPSK (OQPSK) data streams.

pulse streams changes sign, a phase shift of $\pm 90°$ occurs. A change in both streams results in a carrier phase shift of 180°. Figure 7.11a shows a typical QPSK waveform for the sample sequence $d_I(t)$ and $d_Q(t)$ shown in Figure 7.8.

If a QPSK modulated signal undergoes filtering to reduce the spectral sidelobes, the resulting waveform will no longer have a constant envelope and in fact,

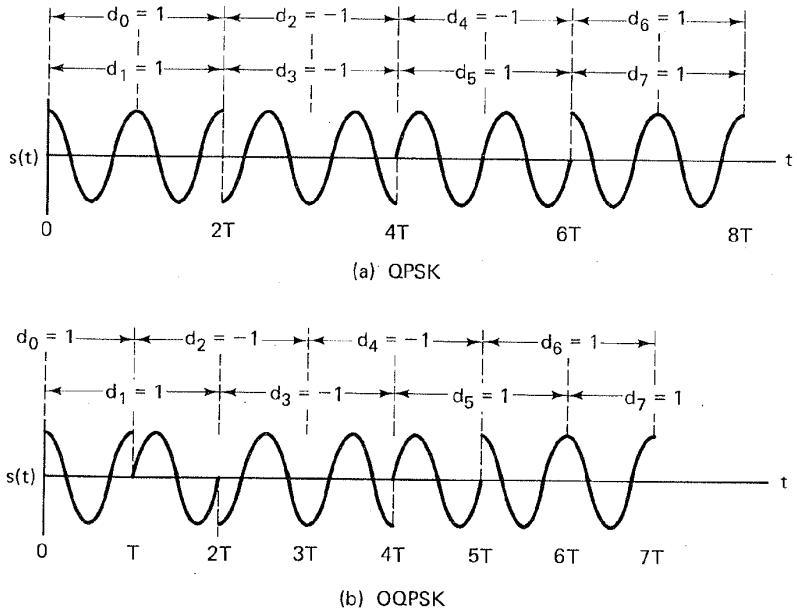


Figure 7.11 (a) QPSK and (b) OQPSK waveforms. (Reprinted with permission from S. Pasupathy, "Minimum Shift Keying: A Spectrally Efficient Modulation," *IEEE Commun. Mag.*, July 1979, Fig. 4, p. 17. © 1979 IEEE.)

the occasional 180° phase shifts will cause the envelope to go to zero momentarily (see Figure 7.11a). When these signals are used in satellite channels employing highly nonlinear amplifiers, the constant envelope will tend to be restored. However, at the same time, all of the *undesirable* frequency side-lobes, which can interfere with nearby channels and other communication systems, are also restored.

In OQPSK, the pulse streams $d_I(t)$ and $d_Q(t)$ are staggered and thus do not change states simultaneously. The possibility of the carrier changing phase by 180° is eliminated, since only one component can make a transition at one time. Changes are limited to 0° and ±90° every T seconds. Figure 7.11b shows a typical OQPSK waveform for the sample sequence in Figure 7.10. When an OQPSK signal undergoes bandlimiting, the resulting intersymbol interference causes the envelope to droop slightly to the region of ±90° phase transition, but since the phase transitions of 180° have been avoided in OQPSK, the envelope will not go to zero as it does with QPSK. When the bandlimited OQPSK goes through a nonlinear transponder, the envelope droop is removed; however, the high-frequency components associated with the collapse of the envelope are not reinforced. Thus out-of-band interference is avoided [7].

7.9.2 Minimum Shift Keying

The main advantage of OQPSK over QPSK, that of suppressing out-of-band interference, suggests that further improvement is possible if the OQPSK format is modified to avoid discontinuous phase transitions. This was the motivation for designing continuous phase modulation (CPM) schemes. *Minimum shift keying* (MSK) is one such scheme [7–9]. MSK can be viewed as either a special case of *continuous-phase frequency shift keying* (CPFSK), or a special case of OQPSK with sinusoidal symbol weighting. When viewed as CPFSK, the MSK waveform can be expressed as [8]

$$s(t) = \cos\left[2\pi\left(f_0 + \frac{d_k}{4T}\right)t + x_k\right] \qquad kT < t < (k+1)T \tag{7.16}$$

where f_0 is the carrier frequency, $d_k = \pm 1$ represents the bipolar data being transmitted at a rate $R = 1/T$, and x_k is a phase constant which is valid over the kth binary data interval. Notice that for $d_k = 1$, the frequency transmitted is $f_0 + 1/4T$, and for $d_k = -1$, the frequency transmitted is $f_0 - 1/4T$. The tone spacing in MSK is thus one-half that employed for noncoherently demodulated orthogonal FSK, giving rise to the name *minimum* shift keying. During each T-second data interval, the value of x_k is a constant, that is, $x_k = 0$ or π, determined by the requirement that the phase of the waveform be continuous at $t = kT$. This requirement results in the following recursive phase constraint for x_k:

$$x_k = \left[x_{k-1} + \frac{\pi k}{2}(d_{k-1} - d_k)\right] \text{ modulo } 2\pi \tag{7.17}$$

Equation (7.16) can be expressed in a quadrature representation, as follows,

using the identities in Equations (D.5) and (D.6):

$$s(t) = a_k \cos \frac{\pi t}{2T} \cos 2\pi f_0 t - b_k \sin \frac{\pi t}{2T} \sin 2\pi f_0 t \qquad kT < t < (k+1)T \quad (7.18)$$

where

$$\begin{aligned} a_k &= \cos x_k = \pm 1 \\ b_k &= d_k \cos x_k = \pm 1 \end{aligned} \quad (7.19)$$

The in-phase (I) component is identified as $a_k \cos (\pi t/2T) \cos 2\pi f_0 t$, where $\cos 2\pi f_0 t$ is the carrier, $\cos (\pi t/2T)$ can be regarded as a *sinusoidal symbol weighting,* and a_k is a data-dependent term. Similarly, the quadrature (Q) component is identified as $b_k \sin (\pi t/2T) \sin 2\pi f_0 t$, where $\sin 2\pi f_0 t$ is the quadrature carrier term, $\sin (\pi t/2T)$ can be regarded as a sinusoidal symbol weighting, and b_k is a data-dependent term. It might appear that the a_k and b_k terms can change every T seconds, since the source data, d_k, can change every T seconds. However, because of the continuous phase constraint, the a_k term can only change value at the zero crossings of $\cos (\pi t/2T)$ and the b_k term can only change value at the zero crossings of $\sin (\pi t/2T)$. Thus the symbol weighting in either the I- or Q-channel is a half-cycle sinsuoidal pulse of duration $2T$ seconds with alternating sign. As in the case of OQPSK, the I and Q components are offset T seconds with respect to one another.

Notice that x_k in Equation (7.17) is a function of the difference between the prior data bit and the present data bit (differential encoding). Hence the a_k and b_k terms in Equation (7.18) can be viewed as *differentially encoded* components of the d_k source data. However, for bit-to-bit independent data d_k, the signs of successive I- or Q-channel pulses are also random from one $2T$-second pulse interval to the next. Thus when viewed as a special case of OQPSK, Equation (7.18) can be rewritten with more straightforward (nondifferential) data encoding [8] as follows:

$$s(t) = d_I(t) \cos \frac{\pi t}{2T} \cos 2\pi f_0 t + d_Q(t) \sin \frac{\pi t}{2T} \sin 2\pi f_0 t \quad (7.20)$$

where $d_I(t)$ and $d_Q(t)$ have the same in-phase and quadrature data stream interpretation as in Equation (7.13). This MSK format in Equation (7.20) is sometimes referred to as *precoded MSK*. Figure 7.12 illustrates Equation (7.20) pictorially. Figure 7.12a and c show the sinusoidal weighting of the I- and Q-channel pulses. These sequences represent the same data sequences as in Figure 7.10, but here, multiplication by a sinusoid results in more gradual phase transitions compared to those of the original data representation. Figure 7.12b and d illustrate the modulation of the orthogonal components $\cos 2\pi f_0 t$ and $\sin 2\pi f_0 t$, respectively, by the sinusoidally shaped data streams. Figure 7.12e illustrates the summation of the orthogonal components from Figure 7.12b and d. In summary, the following properties of MSK modulation can be deduced from Equation (7.20) and Figure

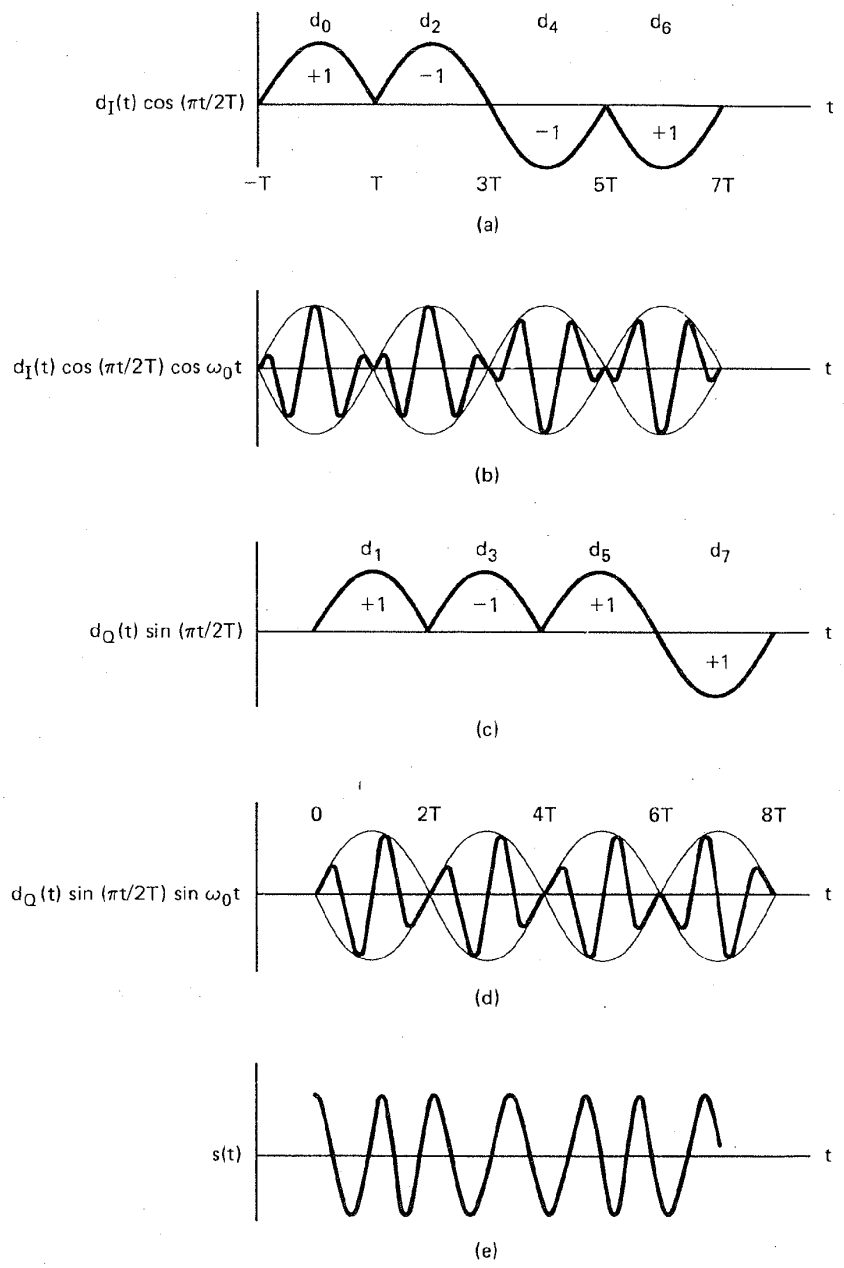


Figure 7.12 Minimum shift keying (MSK). (a) Modified I bit stream. (b) I bit stream times carrier. (c) Modified Q bit stream. (d) Q bit stream times carrier. (e) MSK waveform. (Reprinted with permission from S. Pasupathy, "Minimum Shift Keying: A Spectrally Efficient Modulation," *IEEE Commun. Mag.*, July 1979, Fig. 5, p. 18. © 1979 IEEE.)

7.12: (1) the waveform $s(t)$ has constant envelope; (2) there is phase continuity in the RF carrier at the bit transitions; and (3) the waveform $s(t)$ can be regarded as an FSK waveform with signaling frequencies $f_0 + 1/4T$ and $f_0 - 1/4T$. Therefore, the minimum tone separation required for MSK modulation is

$$\left(f_0 + \frac{1}{4T}\right) - \left(f_0 - \frac{1}{4T}\right) = \frac{1}{2T} \tag{7.21}$$

which is equal to half the bit rate. Notice that the required tone spacing for MSK is one-half the spacing, $1/T$, required for the noncoherent detection of FSK signals (see Section 3.6.4). That is because it is being coherently demodulated.

The power spectral density $G(f)$ for QPSK and OQPSK is given by [8]

$$G(f) = 2PT\left(\frac{\sin 2\pi fT}{2\pi fT}\right)^2 \tag{7.22}$$

where P is the average power in the modulated waveform. For MSK, $G(f)$ is given by [8]

$$G(f) = \frac{16PT}{\pi^2}\left(\frac{\cos 2\pi fT}{1 - 16f^2T^2}\right)^2 \tag{7.23}$$

The normalized power spectral density ($P = 1$ W) for QPSK, OQPSK, and MSK are sketched in Figure 7.13. A spectral plot of BPSK is included for comparison. The fact that BPSK requires more bandwidth than the others for a given level of spectral density should come as no surprise. In Section 7.5.1 and Figure 7.6 we saw that the theoretical bandwidth efficiency of BPSK is half that of QPSK, It is seen from Figure 7.13 that MSK has lower sidelobes than QPSK or OQPSK. This is a consequence of multiplying the data stream with a sinusoid, yielding more *gradual phase transitions*. The more gradual the transition, the faster the spectral tails drop to zero. MSK is *spectrally more efficient* than QPSK or OQPSK; however, as can be seen from Figure 7.13, the MSK spectrum has a wider mainlobe than QPSK and OQPSK. Therefore, MSK may not be the preferred method for narrowband links.

7.9.2.1 Error Performance of OQPSK and MSK

We have seen that BPSK and QPSK have the same bit error probability because QPSK is configured as two BPSK signals modulating orthogonal components of the carrier. Since staggering the bit streams does not change the orthogonality of the carriers, OQPSK has the same theoretical bit error performance as BPSK and QPSK.

Minimum shift keying uses antipodal symbol shapes, $\pm\cos(\pi t/2T)$ and $\pm\sin(\pi t/2T)$, over $2T$ to modulate the two quadrature components of the carrier. Thus when a matched filter is used to recover the data from each of the quadrature components independently, MSK, as defined in Equation (7.20), has the same error performance properties as BPSK, QPSK, and OQPSK [7]. However, if MSK is coherently detected as an FSK signal over an observation interval of T seconds, it would be poorer than BPSK by 3 dB [7]. MSK, with differentially encoded

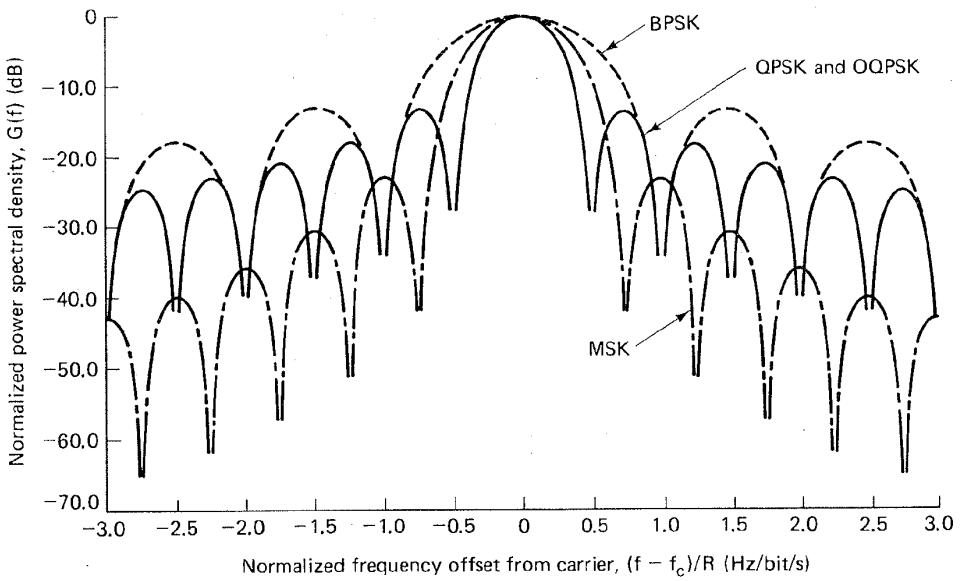


Figure 7.13 Normalized power spectral density for BPSK, QPSK, OQPSK, and MSK. (Reprinted with permission from F. Amoroso, "The Bandwidth of Digital Data Signals," *IEEE Commun. Mag.*, vol. 18, no. 6, Nov. 1980, Fig. 2A, p. 16. © 1980 IEEE.)

data, as defined in Equation (7.16), has the same error probability performance as the coherent detection of differentially encoded PSK.

QPSK systems require a fully coherent or differentially coherent detection scheme. However, since MSK is a type of FSK, it can also be noncoherently detected. This permits inexpensive demodulation of MSK when the value of received E_b/N_0 permits.

7.9.3 Quadrature Amplitude Modulation

Coherent M-ary phase shift keying (MPSK) modulation is a well-known technique for achieving bandwidth reduction. Instead of using a binary alphabet with 1 bit of information per channel symbol period, an alphabet with M symbols is used, permitting the transmission of $k = \log_2 M$ bits during each symbol period. Since the use of M-ary symbols allows a k-fold increase in the data rate within the same bandwidth, then for a fixed data rate, use of M-ary PSK reduces the required bandwidth by a factor k (see Section 3.8.3).

From Equation (7.14) it can be seen that QPSK modulation consists of two independent streams. One stream amplitude-modulates the cosine function of a carrier wave with levels $+1$ and -1, and the other stream similarly amplitude-modulates the sine function. The resultant waveform is termed a double-sideband suppressed-carrier (DSB-SC) wave, since the RF bandwidth is twice the baseband bandwidth (see Section 1.7.1) and there is no isolated carrier term. *Quadrature amplitude modulation* (QAM) can be considered a logical extension of QPSK,

since QAM also consists of two independently amplitude-modulated carriers in quadrature. Each block of k bits (k assumed even) can be split into two ($k/2$)-bit blocks which use ($k/2$)-bit digital-to-analog (D/A) converters to provide the required modulating voltages for the carriers. At the receiver, each of the two signals is independently detected using matched filters. QAM signaling can also be viewed as a combination of amplitude shift keying (ASK) and phase shift keying (PSK), giving rise to the alternative name, *amplitude phase keying* (APK). Finally, it can also be viewed as amplitude shift keying in two dimensions, giving rise to the name *quadrature amplitude shift keying* (QASK).

Figure 7.14a illustrates a two-dimensional signal space and a set of 16-ary QAM signal vectors or points arranged in a rectangular constellation. A canonical QAM modulator is shown in Figure 7.14b. Assuming that Gaussian noise is the only channel disturbance, the simple channel model of Figure 7.14c applies. Signals are sent in pairs (x, y). The model indicates that the signal point coordinates (x, y) are transmitted over separate channels and independently perturbed by Gaussian noise variables (n_x, n_y), each with zero mean and variance N. Or we can say that the two-dimensional signal point is perturbed by a two-dimensional Gaussian noise variable. If the average signal energy (mean-square value of the signal coordinates) is S, then the signal-to-noise ratio is S/N. The simplest method of digital signaling through such a system is to use one-dimensional pulse amplitude modulation (PAM) independently for each signal coordinate. In PAM, to

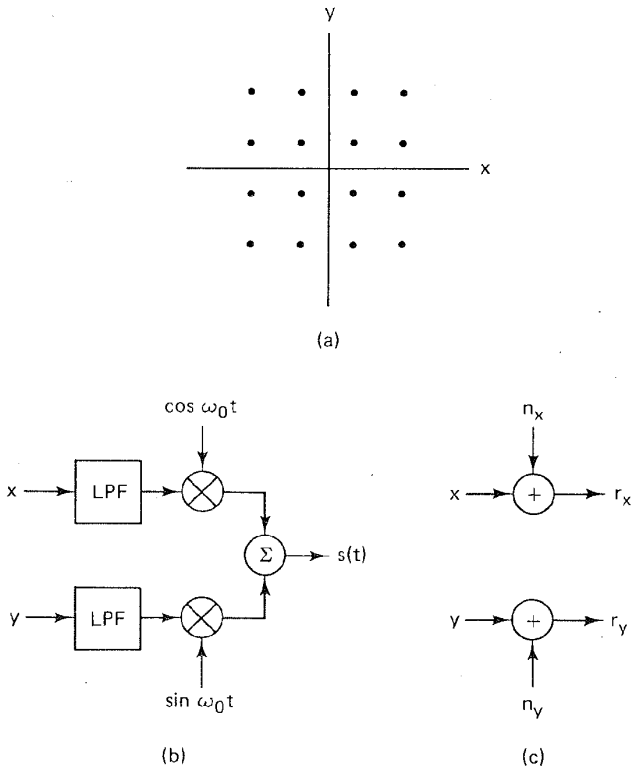


Figure 7.14 QAM modulation. (a) 16-ary signal space. (b) Canonical QAM modulator. (c) QAM channel model.

send k bits/dimension over a Gaussian channel, each signal point coordinate takes on one of 2^k equally likely equispaced amplitudes. By convention, the signal points are grouped about the center of the space at amplitudes $\pm 1, \pm 3, \ldots, \pm(2^k - 1)$.

7.9.3.1 QAM Probability of Bit Error

For a rectangular constellation, a Gaussian channel, and matched filter reception, the probability of bit error is expressed [10] by

$$P_B \simeq \frac{2(1 - L^{-1})}{\log_2 L} Q\left[\sqrt{\left(\frac{3 \log_2 L}{L^2 - 1}\right)\frac{2E_b}{N_0}}\right] \tag{7.24}$$

where $Q(x)$ is as defined in Equation (2.42) and L represents the number of amplitude levels in one dimension. We assume that a sequence of $\log_2 L$ bits are assigned to an L-ary symbol using a Gray code (defined in Section 3.9.4).

7.9.3.2 Bandwidth–Power Trade-Off

The bandwidth–power trade-off of M-ary QAM at a bit error probability of 10^{-5} is displayed on the bandwidth-efficiency plane in Figure 7.6, with the abscissa measured in average E_b/N_0. We assume Nyquist filtering of the baseband pulses so that the DSB transmission bandwidth at IF is $W_{IF} = 1/T$, where T is the symbol duration. Thus the bandwidth efficiency is $R/W = \log_2 M$, where M is the symbol set size. For realistic channels and waveforms, the performance must be reduced to account for the increased bandwidth necessary to implement realizable filters. From Figure 7.6 it can be seen that QAM represents a method of reducing the bandwidth required for the transmission of digital data. As with M-ary PSK, bandwidth efficiency can be exchanged for power or E_b/N_0; however, in the case of QAM, a *much more efficient exchange* is possible than in the case of M-ary PSK.

For a comparative treatment of digital modulation techniques in general, Reference [11] contains useful performance data and an extensive list of other references.

Example 7.3 Waveform Design

Assume that a data stream with data rate $R = 144$ Mbits/s is to be transmitted on an RF channel using a DSB modulation scheme. Assume Nyquist filtering and an allowable DSB bandwidth of 36 MHz. Which modulation technique would you choose for this requirement? If the available E_b/N_0 is 20, what would be the resulting probability of bit error?

Solution

The required spectral efficiency is

$$\frac{R}{W} = \frac{144 \text{ Mbits/s}}{36 \text{ MHz}} = 4 \text{ bits/s/Hz}$$

From Figure 7.6 we note that 16-ary QAM, with a theoretical spectral efficiency of 4 bits/s/Hz, requires a lower E_b/N_0 than that of 16-ary PSK for the same P_B. Based on these considerations we choose a 16-ary QAM modem.

With the available E_b/N_0 given as 20, we use Equation (7.24) to calculate the expected bit error probability as

$$P_B \simeq \frac{3}{4} Q\left(\sqrt{\frac{4}{5}\frac{E_b}{N_0}}\right) = 2.5 \times 10^{-5}$$

Example 7.4 Spectral Efficiency

(a) Explain the computation of the QAM spectral efficiency in Example 7.3, considering that QAM is transmitted on orthogonal components of a carrier wave.
(b) Since the DSB bandwidth is 36 MHz in Example 7.3, consider using half that amount at baseband to transmit the 144-Mbits/s data stream, using multilevel PAM. What is the spectral efficiency needed to accomplish this, and how many levels of PAM would be required? Assume Nyquist filtering.

Solution

(a) *Bandpass channel using QAM:* The 144-Mbits/s data stream is partitioned into a 72-Mbits/s in-phase and a 72-Mbits/s quadrature stream; one stream amplitude-modulates the cosine component of a carrier over a bandwidth of 36 MHz, and the other stream amplitude-modulates the sine component of the carrier wave over the same 36-MHz bandwidth. Since each 72-Mbits/s stream modulates an orthogonal component of the carrier, the 36 MHz suffices for both streams, or for the full 144 Mbits/s. Thus the spectral efficiency is (144 Mbits/s)/36 MHz = 4 bits/s/Hz.
(b) *Required spectral efficiency at baseband*

$$\frac{R}{W} = \frac{144 \text{ Mbits/s}}{18 \text{ MHz}} = 8 \text{ bits/s/Hz}$$

Assuming Nyquist filtering, a bandwidth of 18 MHz can support a maximum symbol rate of $R_s = 2W = 36$ megasymbols/s [see Equation (2.76)]. Each PAM pulse must therefore have an ℓ-bit meaning, such that

$$R = \ell R_s$$

Hence

$$\ell = \frac{144 \text{ Mbits/s}}{36 \text{ megapulses/s}} = 4 \text{ bits/pulse}$$

where $\ell = \log_2 L$, and $L = 16$ levels.

7.10 MODULATION AND CODING FOR BANDLIMITED CHANNELS

The channel coding techniques of Chapters 5 and 6 have generally *not* been associated with voice-grade telephone channels (although the first field test of sequential decoding of convolutional codes was on a telephone line). Recently, however, there has been considerable interest in techniques that can provide coding gain for bandlimited channels. The motivation is to enable the reliable transmission of *higher data rates* over voice-grade channels. The potential gain

is about 3 bits/symbol (for a given signal-to-noise ratio) [12] or, alternatively, a given error performance could be achieved with a power savings of 9 dB [12].

The greatest interest is in the following three separate coding research areas:

1. Optimum signal constellation boundaries (choosing a closely packed signal subset from any regular array or lattice of candidate points)
2. Higher-density lattice structures (adding improvement to the signal subset choice by starting with the densest possible lattice for the space)
3. Trellis-coded modulation (combined modulation and coding techniques for obtaining coding gain for bandlimited channels)

The first two areas are not "true" error control coding schemes. By "true error control coding" we refer to those techniques that employ some structured redundancy to improve the error performance. Only the third technique, trellis-coded modulation, involves redundancy. Each of these coding research areas and their expected performance improvements are discussed below.

7.10.1 Commercial Telephone Modems

The use of efficient modulation techniques has traditionally been spearheaded by the telecommunications industry, since the telephone company's foremost resource consists of sharply bandlimited voice-grade channels. The typical telephone channel is characterized by a high signal-to-noise ratio (SNR) of approximately 30 dB and a bandwidth of approximately 3 kHz. Table 7.1 lists the evolution of high-speed telephone modems with bandwidth efficiencies (R/W) ranging from 2 to 8 bits/s/Hz. The list starts with the Bell 201, introduced in about 1962, which used QPSK in a nominal 1200-Hz bandwidth to achieve 2400 bits/s on private lines. The first commercially important 4800-bits/s modem was the Milgo 4400/48, introduced in about 1967. It utilized a nominal 1600-Hz bandwidth in conjunction with 8-ary PSK to achieve a bandwidth efficiency of 3 bits/s/Hz. In 1971 the Codex 9600C was introduced. It provided 9600 bits/s in a 2400-Hz

TABLE 7.1 Modem Milestones

Year	Model	Speed (bits/s)	Bandwidth (Hz)	Modulation	R/W (bits/s/Hz)
1962	Bell 201	2,400	1200	4-PSK	2
1967	Milgo 4400/48	4,800	1600	8-PSK	3
1971	Codex 9600C	9,600	2400	16-QAM	4
1980	Paradyne MP14400	14,400	2400	64-QAM	6
1981	Codex SP14.4	14,400	2400	64-QAM	6
1984	Codex 2660	14,400	2400	Trellis-coded QAM	6
1985	Codex 2680	19,200	2400	Trellis-coded QAM	8

bandwidth (R/W = 4 bits/s/Hz) using 16-ary QAM. Note that as channel equalization techniques (see Section 2.11.2) improved, a larger bandwidth portion of the voice-grade channel became usable. Whereas in 1962, only 1200 Hz could be reliably employed, that value doubled by 1971.

In 1980, first-generation 14,400-bits/s modems were introduced by Paradyne (MP14400), followed in 1981 by Codex (SP14.4). These modems improved the bandwidth efficiency by utilizing 64-ary QAM with an R/W of 6 bits/s/Hz. In a second generation, appearing in 1984, trellis-coded QAM modulation (treated in Section 7.10.6) was introduced to provide better error performance. In 1985, Codex introduced a modem with R/W = 8 bits/s/Hz, thereby achieving a data rate of 19,200 bits/s in a nominal bandwidth of 2400 Hz. Without any major upgrading of the telephone network, 19,200 bits/s is considered the maximum achievable data rate for an unconditioned voice-grade telephone channel [12].

7.10.2 Signal Constellation Boundaries

Several researchers [13–17] have examined large numbers of possible QAM signal constellations in a search for designs that result in the best error performance for a given average signal-to-noise ratio. Figure 7.15 illustrates some examples of symbol constellations for M = 4, 8, and 16 that have been considered [13]. The circular sets are designated by the notation (a, b, . . .), where there are a quantity of a signals on the inner circle, b signals on the next circle, and so on. In general, the constellation rule, known as the Campopiano–Glazer construction rule [15], that yields optimum signal set performance can be summarized as follows: From an infinite array of points closely packed in a *regular array or lattice,* select a closely packed subset of 2^k points as a signal constellation. In this case "optimum" means minimum average or peak power for a given error probability. In a two-dimensional signal space the optimum boundary surrounding an array of points tends toward a circle. Figure 7.16 illustrates examples of 64-ary (k = 6) and 128-ary (k = 7) signal sets from a rectangular array. The cross-shaped boundaries are a compromise to the optimum circle. The k = 6 constellation was used in the Paradyne 14.4-kbits/s modem. Compared to a square, the performance improvement resulting from a circular boundary is only a modest 0.2 dB [12].

7.10.3 Higher-Dimensional Signal Constellations

For any particular information rate and a channel noise process that is independent and identically distributed in the two dimensions, signaling in a two-dimensional QAM space can provide the same error performance with less average (or peak) power than signaling in a one-dimensional pulse amplitude (PAM) space. We stated earlier that this is accomplished by choosing signal points on a two-dimensional lattice from within a circular rather than a rectangular boundary. In the same way, by going to a higher number, N, of dimensions and choosing points on an N-dimensional lattice from within an N-sphere rather than an N-cube, further energy savings are possible. Several researchers [18–21] have studied multidimensional signal constellations. Consider the four-dimensional configuration

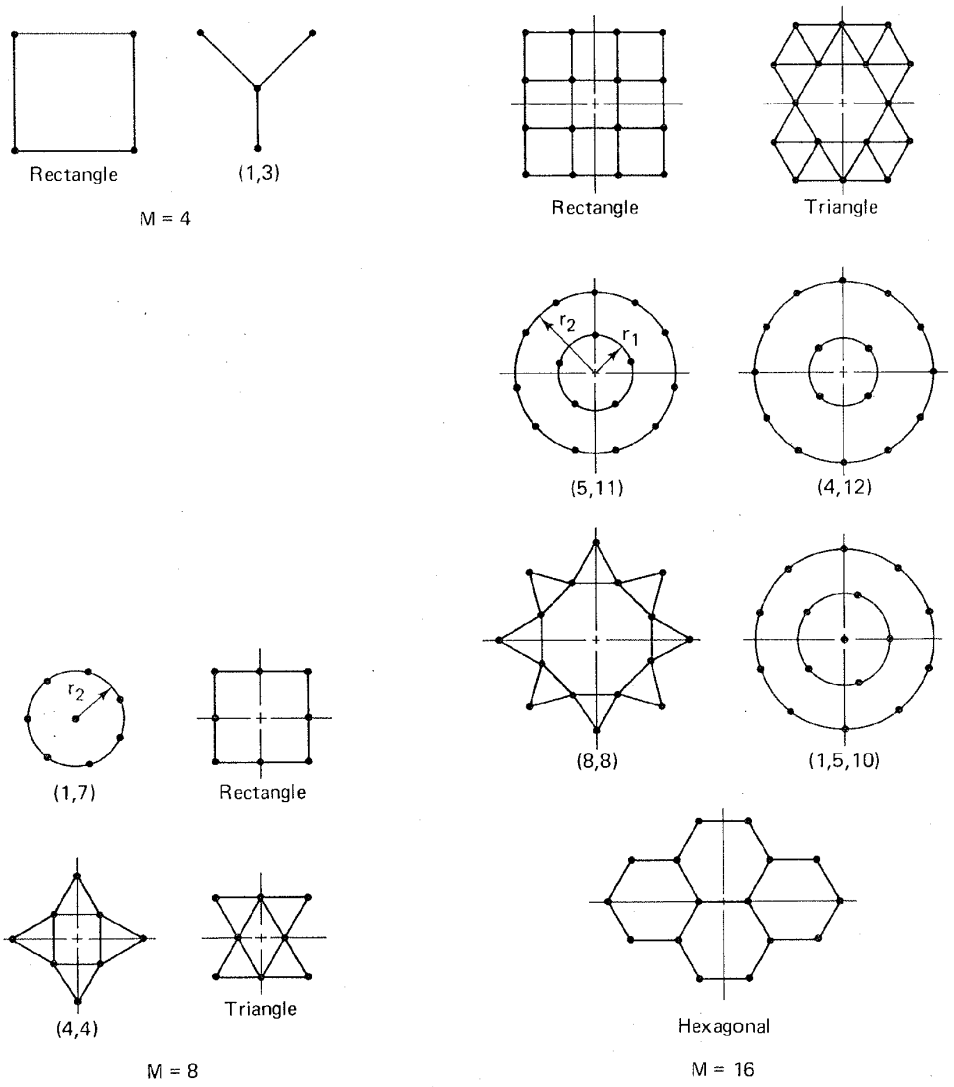


Figure 7.15 *M*-ary symbol constellations. (Reprinted with permission from C. M. Thomas, M. Y. Weidner, and S. H. Durrani, "Digital Amplitude-Phase Keying with *M*-ary Alphabets," *IEEE Trans. Commun.*, vol. COM22, no. 2, Feb. 1974, Figs. 2 and 3, p. 170. © 1974 IEEE.)

illustrated in Figure 7.17. The transmitter transmits four simultaneous sequences of pulses over four bandlimited Gaussian channels. We assume that the source produces one of M symbols, $m_i = 1, 2, \ldots, M$, every T seconds. A given symbol m_i causes four pulses to be emitted—$a_i s(t)$, $b_i s(t)$, $c_i s(t)$, $d_i s(t)$—as shown in Figure 7.17. These are transmitted on separate noninterfering channels. The pulses are distorted by independent AWGN in each channel, and at the receiver they are detected separately with matched filters. The four independent channels can

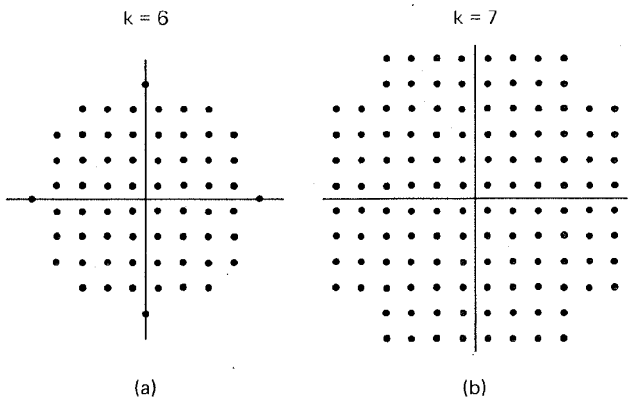


Figure 7.16 Examples of M-ary constellations using a rectangular array.

be implemented in a number of ways:

1. Two bandpass channels can be used, each with separately modulated in-phase and quadrature components (QAM or MPSK modulation on each channel).
2. The two bandpass channels can be time- or frequency-division multiplexed and carried on a common transmission line.
3. Orthogonal electromagnetic wave polarization can be used.

Let us compare a two-dimensional 16-ary QAM system with a four-dimensional alternative. In the two-dimensional modulation case, during each T-second

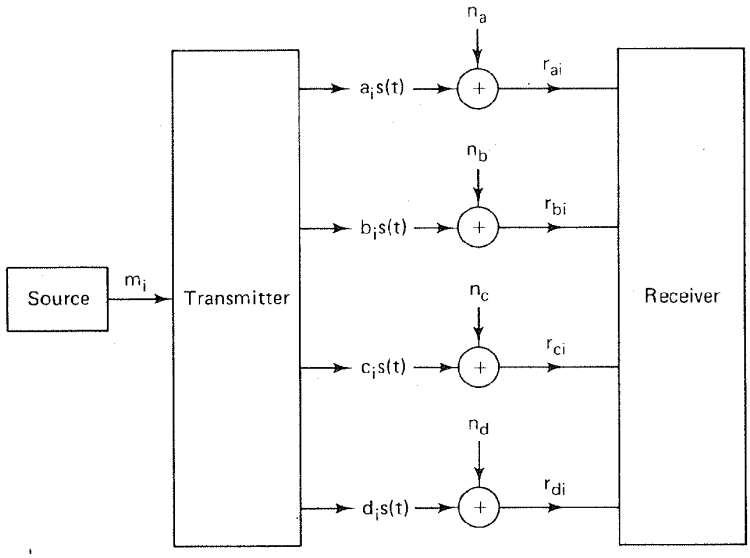


Figure 7.17 Four-dimensional system configuration.

interval, a symbol (4 bits) is transmitted by the modulation of the in-phase and quadrature components of a carrier. In other words, two 4-ary numbers can be transmitted to describe a signal vector in the 16-ary two-dimensional signaling plane. In the case of four-dimensional modulation, two successive symbols (8 bits) are sent each $2T$ seconds by transmitting four 4-ary numbers representing a point in a 256-ary signal space. It can be shown that increasing the dimensionality of the signal space offers a potential savings in average signal energy for a given level of error performance. That is, in going to a higher-dimensional space, one can effect an energy savings based on the selection of signal points from an N-sphere versus an N-cube of the same volume—the average energy of the signal points from the N-sphere is less than that from the N-cube. Table 7.2 gives the energy savings possible in N dimensions. Of course, the implementation of such a scheme involves added complexity. To send n bits per symbol in N dimensions (assuming N even), incoming bits must be grouped in blocks of $nN/2$. A mapping must then be performed into the space of $2^{(nN/2)}$ N-dimensional vectors which have the least energy among all such vectors. A corresponding inverse mapping must be made at the receiver. The added complexity may, of course, outweigh the performance gain. As N goes to infinity, the gain goes to 1.53 dB [12].

TABLE 7.2 Energy Savings from N-Sphere Mapping versus N-Cube Mapping

Dimensions (N)	N-sphere mapping gain (dB)
2	0.20
4	0.45
8	0.73
16	0.98
24	1.10
32	1.17
48	1.26
64	1.31

Source: G. D. Forney, Jr., et al., "Efficient Modulation for Bandlimited Channels," *IEEE J. Sel. Areas Commun.*, vol. SAC2, no. 5, September 1984, pp. 632–647.

7.10.4 Higher-Density Lattice Structures

In Section 7.10.3 we discussed the selection of a closely packed subset of points from any regular array or lattice. Here we consider the added improvement by starting with the *densest possible lattice* in the space. In a two-dimensional signal space, the densest lattice is the hexagonal lattice (try penny packing). The result of employing a hexagonal lattice instead of a rectangular one, such as those shown in Figure 7.16, can be a 0.6-dB savings in average energy. Figure 7.18 illustrates some examples of hexagonal packing. The strange-looking $k = 4$ constellation in Figure 7.18a was discovered by Foschini et al. [17] and is stilll the best 16-ary

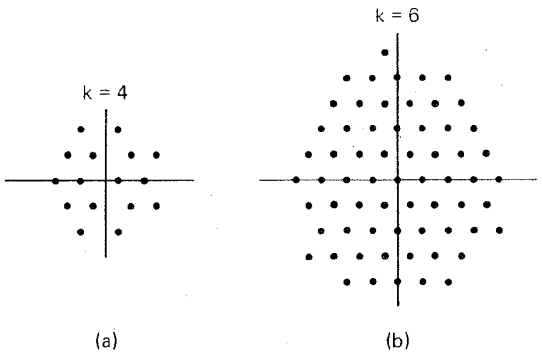


Figure 7.18 Examples of M-ary constellations using a hexagonal array.

constellation known. The $k = 6$ constellation in Figure 7.18b is used in the Codex SP14.4 modem.

The hexagonal lattice is optimum for two dimensions. For higher dimensions there are other lattice structures that provide the densest packing. Table 7.3 gives the gain over the rectangular lattice, in decibels, due to the densest packings currently known for various dimensions.

TABLE 7.3 Energy Savings from Dense Lattices versus the Rectangular Lattice

Dimensions (N)	Dense lattice gain (dB)
2	0.62
4	1.51
8	3.01
16	4.52
24	6.02
32	6.02
48	7.78
64	8.09

Source: G. D. Forney, Jr., et al., "Efficient Modulation for Bandlimited Channels," *IEEE J. Sel. Areas Commun.*, vol. SAC2, no. 5, September 1984, pp. 632–647.

7.10.5 Combined Gain: *N*-Sphere Mapping and Dense Lattice

It is possible to combine the benefits of the Campopiano–Glazer boundary construction in N dimensions with the gain from the densest lattice in N-space. The resulting gain is a combination of N-sphere versus N-cube boundary gain of Table 7.2 and the lattice packing density gain of Table 7.3. The combined energy savings are shown in Table 7.4.

TABLE 7.4 Combined Energy Savings from N-Sphere Mapping and Dense Lattices

Dimensions (N)	Combined savings gain (dB)
2	0.82
4	1.96
8	3.74
16	5.50
24	7.12
32	7.19
48	9.04
64	9.40

Source: G. D. Forney, Jr., et al., "Efficient Modulation for Bandlimited Channels," *IEEE J. Sel. Areas Commun.*, vol. SAC2, no. 5, September 1984, pp. 632–647.

7.10.6 Trellis-Coded Modulation

The codes described in Chapters 5 and 6 achieve an improvement in bit error probability (P_B) by *bandwidth expansion*. In the case of both block codes and convolutional codes, bandwidth is increased by replacing each k-tuple message with an n-tuple codeword, where $n > k$. In the case of bandlimited channels, *bandwidth expansion is not possible*. In the past, therefore, coding has never been popular for bandlimited channels such as telephone channels. Recently, however, there has been increasing interest in some types of combined modulation and coding schemes, called *trellis-coded modulation,* that achieve coding gain without any bandwidth expansion. At first it may seem that this statement violates some basic power–bandwidth–error probability trade-off principle. However, there is still a trade-off at work; trellis-coded modulation achieves coding gain at the expense of *decoder complexity*.

Trellis-coded modulation combines a *multilevel/phase* modulation signaling set with a state-oriented trellis coding scheme. Multilevel/phase signal sets are signal constellations having multiple amplitudes, multiple phases, or a combination of multiple amplitudes and multiple phases. A trellis code is one that can be characterized with a trellis diagram. The convolutional codes described in Chapter 6 are linear trellis codes, but trellis codes are *not constrained to be linear*. Coding gains can be realized with block codes or trellis codes, but we shall consider only trellis codes because the availability of the Viterbi decoding algorithm makes trellis decoding attractive. Coding for bandlimited channels still requires controlled introduction of redundancy. However, in this case, the redundancy is due to an increased signal alphabet, achieved through multilevel/phase signaling, so that channel bandwidth is not increased. Ungerboeck [22] investigated the design of multilevel/phase trellis codes that provide *coding gain without band-*

width expansion. He showed that in the presence of AWGN, net coding gains of 3 to 6 dB, relative to the uncoded case, could be achieved.

7.10.6.1 The Idea behind Trellis-Coded Modulation

The error performance of an uncoded nonorthogonal M-ary modulation (such as PAM, PSK, or QAM) depends on the distance between the closest pair of signal points. This minimum distance is determined by the average transmitter power and the number and position of the signal points. For a constant average power, the minimum distance between points decreases as the number of points increases. Therefore, assuming a constant channel symbol rate and constant average power, the error performance degrades for systems that attempt to increase the transmission bit rate by increasing the size of the symbol set. The objective of trellis coding is to increase the minimum distance between the signals that are the *most likely to be confused,* without increasing the average power.

Trellis coding may be implemented with a convolutional encoder (see Chapter 6) wherein k current bits and $K - 1$ prior bits are used to produce $n = k + p$ coded bits, where K is the encoder constraint length and where p is the number of parity bits. The $n = k + p$ coded bits require 2^n binary channel symbols for transmission. Notice that encoding increases the signal set size from 2^k to 2^{k+p}. Figure 7.19a illustrates an uncoded 4-ary PAM signal set, before and after being rate $\frac{2}{3}$ encoded into an 8-ary PAM signal set. Similarly, Figure 7.19b illustrates an uncoded 4-ary PSK (QPSK) signal set before and after being rate $\frac{2}{3}$ encoded into an 8-ary PSK signal set. Similarly, Figure 7.19c illustrates an uncoded 16-

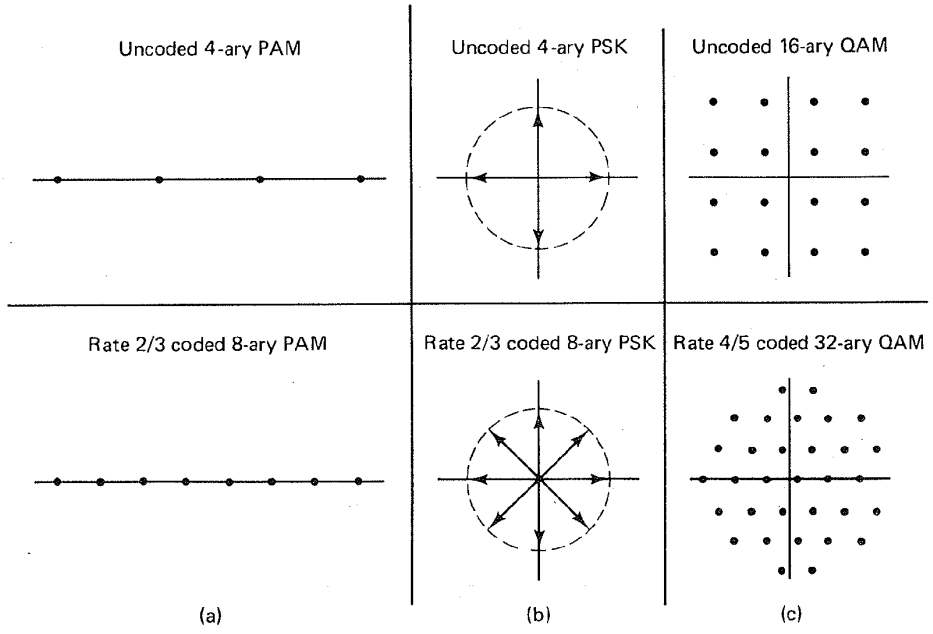


Figure 7.19 Increase of signal set size for trellis-coded modulation.

ary QAM signal set before and after being rate $\frac{4}{5}$ encoded into a 32-ary QAM signal set. In each of these three cases the system is configured to use the same average signal power before and after coding. The examples in Figure 7.19 illustrate the basic idea behind trellis-coded modulation. In each case the symbol set size is increased from 2^k to 2^{k+1} (there are twice as many coded symbols as uncoded ones) to provide the needed coding redundancy; however, in each case, the increase in the number of signals *does not* result in an increase in required bandwidth. The expanded signal set does result in a *reduced distance* between adjacent symbol points for a given average power. However, because of the redundancy introduced by the code, this reduced distance no longer determines the error performance. Instead, the *free distance* (see Section 6.4.1), which is the minimum distance between members of the set of *allowable code symbol sequences,* determines the error performance. Ungerboeck [22] investigated the increase in channel capacity achievable by signal set expansion and concluded that by *doubling* the number of channel signals ($p = 1$), it is possible to gain almost all the channel capacity that can be gained. This can be accomplished by encoding with a rate $k/(k + 1)$ code, and subsequently mapping groups of $k + 1$ bits into the larger set of 2^{k+1} channel symbols.

7.10.6.2 An Error Event

Figure 7.20 illustrates an error event in a trellis code; that is, the figure illustrates a transmitted sequence marked $\mathbf{U} = \ldots, U_1, U_2, U_3, \ldots$ and an alternative sequence marked $\mathbf{V} = \ldots, V_1, V_2, V_3, \ldots$. The alternative sequence is

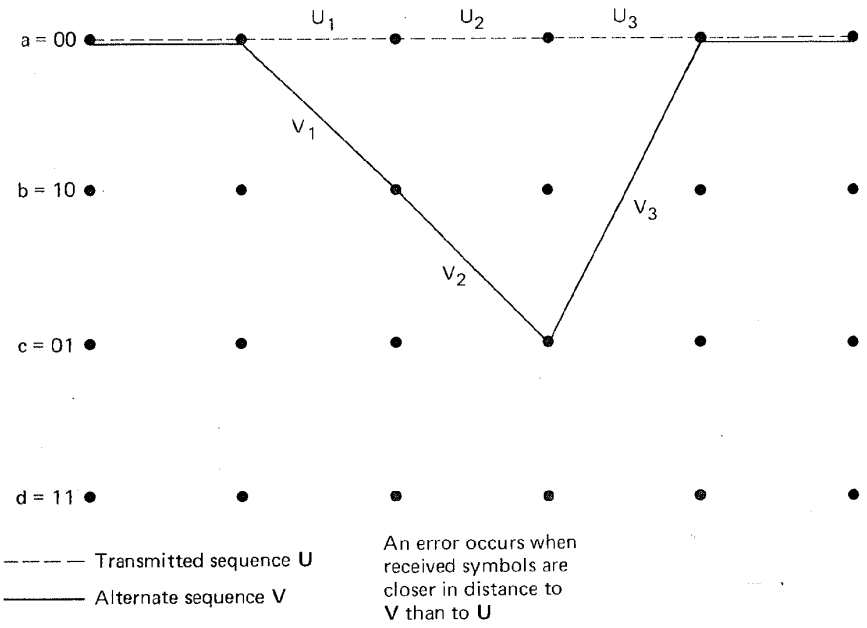


Figure 7.20 Illustration of an error event.

seen to diverge and then remerge with the transmitted sequence. Assuming soft-decision decoding, an *error event occurs* whenever the received symbols are closer in Euclidean distance to some alternative sequence, **V**, than to the actually transmitted sequence, **U**. Thus the separation of **U** and **V** describes an error event. This implies that codes for multilevel/phase signals should be designed to achieve *maximum free Euclidean distance* rather than maximum free Hamming distance; for soft-decision decoding, the larger the Euclidean distance, the lower the probability of error. Therefore, assigning signal points to the coded bits in a way that maximizes Euclidean distance is the key to optimizing the trellis codes. Ungerboeck [22–24] investigated this bit-to-symbol mapping problem and devised an assignment procedure, called the *method of set partitioning,* which will always provide coding gain, given an adequate choice of trellis states. The rules for this bit-to-symbol mapping are based on the method of set partitioning, which can be summarized as follows:

1. All parallel transitions in the trellis structure are separated by the maximum possible Euclidean distance. *Parallel transitions* refer to the branch words resulting from the transmission of uncoded bits together with coded bits (see the example in the following section). The reasoning behind this is based on the fact that parallel transitions imply that single signal-error events can occur. This limits the achievable free Euclidean distance to the minimum distance in the subsets of signals assigned to parallel transitions.
2. All transitions diverging from or merging into a trellis state are assigned the next maximum possible Euclidean distance separation.

In summary, trellis coding for bandlimited channels employs larger signal alphabets achieved through multilevel/phase signaling, such that channel bandwidth is not increased (e.g., M-ary PAM, MPSK, or QAM). Even though the increase in signal set size *reduces* the minimum distance between symbols, the free Euclidean distance between trellis code sequences *more than compensates* for the signal points being crowded together. The result is a net error-performance gain of 3 to 6 dB without any bandwidth expansion [22]. We illustrate these ideas by considering a rate $\frac{2}{3}$ convolutional encoder in the following section.

7.10.7 Trellis-Coding Example

A rate $\frac{2}{3}$ convolutional encoder with constraint length $K = 3$ is shown in Figure 7.21. The rate $\frac{2}{3}$ encoding is accomplished by transmitting one bit from each pair of bits in the input sequence unmodified, and encoding the other bit into two channel bits using a rate $\frac{1}{2}$ encoder. The resulting trellis diagram is shown in Figure 7.22, where the parallel transitions are due to the uncoded bit m_1 shown as the leftmost bit on each trellis branch. The two upper branches emerging from each state represent transitions due to m_1m_2 being 00 and 10, respectively; the two lower branches represent transitions due to m_1m_2 being 01 and 11, respectively. The Viterbi decoding technique for finding the maximum likelihood path through the trellis proceeds in exactly the same way as in the example of Section 6.3.4.

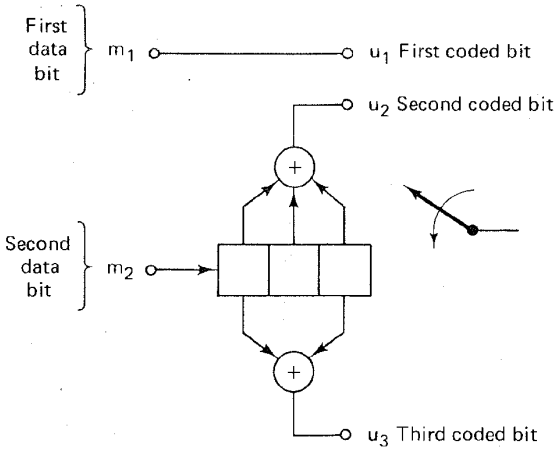


Figure 7.21 Rate $\frac{2}{3}$ convolutional encoder.

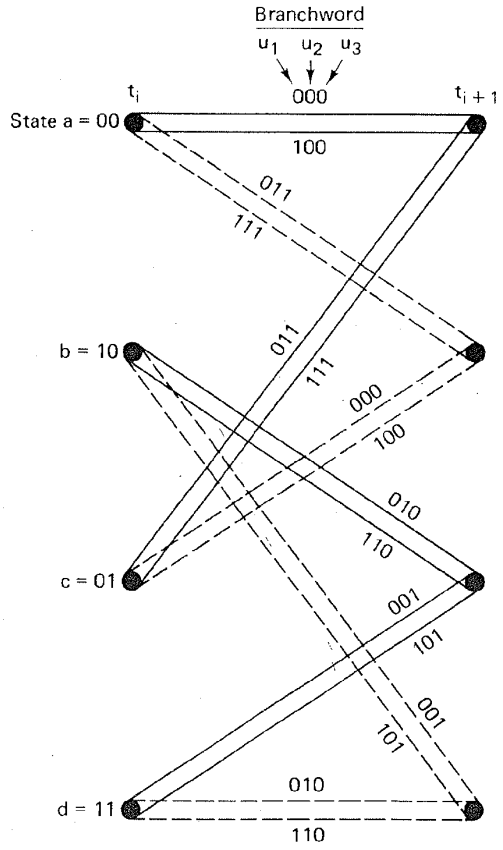


Figure 7.22 Trellis diagram (rate $\frac{2}{3}$ code).

The only (operational) differences are: (1) In this rate $\frac{2}{3}$ example, there are twice as many branches to consider than in the ordinary $K = 3$ convolutional code; and (2) in choosing the two decoded bits for a surviving branch, the first decoded

bit of the pair is the same as the first bit u_1 of that branch word since u_1 is the same as the *uncoded* bit m_1. The second decoded bit of the pair corresponds to the input bit m_2 that produced the state transition of the branch being decoded. In Figure 7.22 a branch having a solid line corresponds to $m_2 = 0$, and a branch having a dashed line corresponds to $m_2 = 1$.

7.10.7.1 Coding Gain for Trellis Coding

Consider the coding gain of the rate $\frac{2}{3}$ trellis-coding example described in the preceding section. Let us assume a simple one-dimensional signal space with multilevel pulse amplitude modulation (PAM), as shown in Figure 7.23. In Figure 7.23a is an 8-ary PAM signal set. Since soft decisions are assumed, the appropriate distance metric is the Euclidean distance. The Euclidean distance of each signal, from the center of the signal space, is shown in arbitrary units. Also shown in Figure 7.23a is the bit-to-symbol assignment according to the set partitioning rules outlined earlier. Notice the adherence to these rules, by comparing Figure 7.23a with Figure 7.22. All parallel transitions are separated by a distance of eight units, and all branches diverging from a given state are separated by at least four units.

The average signal power, S_{av}, is computed as follows:

$$S_{av} = \frac{d_1^2 + d_2^2 + \cdots + d_M^2}{M} \tag{7.25}$$

where d_i is the Euclidean distance of the ith signal from the center of the space, and M is the number of codeword symbols in the set. For the signal set shown in Figure 7.23a, where $M = 8$, use of Equation (7.25) yields $S_{av} = 21$. Figure 7.23b illustrates a 4-ary PAM signal set which is the uncoded equivalent of the rate $\frac{2}{3}$ codeword set; the Euclidean distances have been chosen to yield the same average signal power as in the coded case in Figure 7.23a.

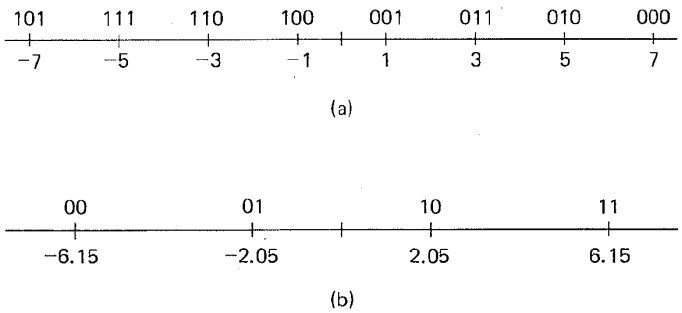


Figure 7.23 8-ary and 4-ary PAM signal sets.

Figure 7.24 illustrates the minimum distance error event for the rate $\frac{2}{3}$ encoder shown in Figure 7.21. The transmitted sequence is assumed to correspond to the all-zeros path. Each of the branch words on this path has a Euclidean distance of 7 units from the center of the space. The error event diverges from the all-zeros path by first transitioning to state 10, then state 01, and finally re-

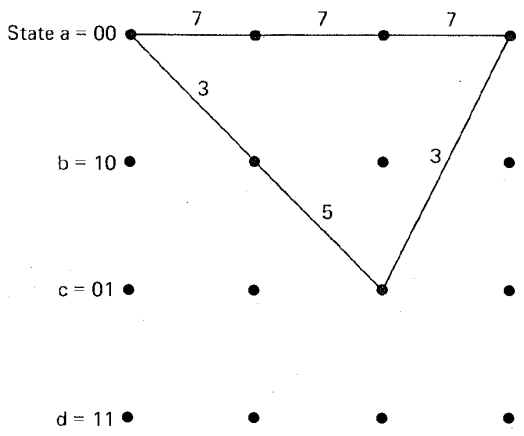


Figure 7.24 Minimum distance error event for rate $\frac{2}{3}$ convolutional code.

merging to state 00. On each branch of the error event is written its Euclidean distance from the center of the space, assuming the uncoded bit $m_1 = 0$; this assumption assures that the distance between the all-zeros transmission and the error event is minimum. To verify the Euclidean distances for the error event in Figure 7.24, first note the state from which and to which each branch transitions. From Figure 7.22 determine the branch word sequence for each transition (with $m_1 = 0$), and then from Figure 7.23a determine the Euclidean distance from the center of the space for each branch in the error event.

Signal amplitude corresponds to the distance of the signal point from the center of the signal space; signal power corresponds to the square of this distance. In comparing the relative performance of the uncoded 4-ary PAM modulation with the trellis-coded 8-ary modulation, it is therefore appropriate to compare the square of the minimum distance d^2_{min} for an error event in each system, given that the average power is the same in both cases. In general, allowing the average signal power to be different in each case, we can solve for the coding gain G, as follows [25]:

$$G = \frac{(d^2_{\text{min}}/S_{\text{av}})_{\text{coded}}}{(d^2_{\text{min}}/S_{\text{av}})_{\text{uncoded}}} \tag{7.26}$$

For the rate $\frac{2}{3}$ trellis-coding example, the d^2_{min} value for each error event is calculated as follows:

$$(d^2_{\text{min}})_{\text{coded}} = (7 - 3)^2 + (7 - 5)^2 + (7 - 3)^2 = 36$$

$$(d^2_{\text{min}})_{\text{uncoded}} = (6.15 - 2.05)^2 = 16.8$$

Then the coding gain is calculated by using Equation (7.26) as follows:

$$G = \frac{36/21}{16.8/21} = 2.14$$

$$\text{or, in decibels} = 3.31 \text{ dB}$$

Therefore, even for this simple $K = 3$ example, a significant amount of coding

gain has been provided *without any bandwidth expansion*. Larger coding gains can be achieved with an increased number of trellis states (larger constraint length) at the expense of increased decoding complexity. Table 7.5 lists the average power coding gain as a function of number of trellis states [26] for the rate $\frac{2}{3}$ coded 8-ary PAM example discussed here. Gain is computed relative to an uncoded 4-ary PAM signal set.

TABLE 7.5 Coding Gain Obtained for 8-ary PAM with Rate 2/3 Trellis Coding

Number of trellis states	Constraint length (K)	Average power gain (dB)
4	3	3.31
8	4	3.77
16	5	4.18
32	6	4.56
64	7	5.23
128	8	5.23
256	9	5.83

Source: G. C. Clark, Jr. and J. B. Cain, *Error Correction Coding for Digital Communications*, Plenum Press, New York, 1981, p. 388.

At the transmitter, there is only a slight increase in complexity due to the trellis coding. However, the decoding problem at the receiver is made much more complex [12], so that the trade-off consists of evaluating the *coding gain versus the decoding complexity*. The increased availability of large-scale integrated (LSI) circuits and very high speed integrated circuits (VHSIC) can ameliorate this problem and make these coding techniques extremely attractive for achieving coding gain for bandlimited channels. Before concluding, we point out that further coding gain, without bandwidth expansion, is possible by introducing asymmetry into the signal point constellation [27–28].

7.11 CONCLUSION

In this chapter we have integrated some of the ideas in Chapters 3, 5, and 6 dealing with modulation and coding. We have reviewed the basic system design goals: to maximize data rate while simultaneously minimizing error probability, bandwidth, E_b/N_0, and complexity. We examined the trade-offs heuristically on two performance planes: the error probability plane and the bandwidth efficiency plane. The former explicitly illustrates the P_B versus E_b/N_0 trade-offs while only implicitly displaying the bandwidth expenditure. The latter explicitly illustrates the R/W versus E_b/N_0 trade-offs while only implicitly displaying the P_B performance. We discussed some of the basic constraints to improvement without limit. The Nyquist criterion establishes that we cannot continue to reduce system bandwidth indefinitely. There is a theoretical limitation; in order to transmit R_s symbols/second without intersymbol interference, we must utilize a minimum of $R_s/2$ hertz of bandwidth. The Shannon–Hartley theorem relates to the power–band-

width trade-off and results in another important limitation, the Shannon limit. The Shannon limit of -1.59 dB is the minimum amount of E_b/N_0 that is necessary (in concert with channel coding) to achieve an arbitrarily low error probability over an AWGN channel. The more general limitation is the channel capacity, above which there cannot be error-free signaling. We have also examined some of the bandwidth-efficient modulation schemes, such as minimum shift keying (MSK), quadrature amplitude modulation (QAM), and trellis-coded modulation. The latter technique offers an attractive way to obtain coding gain without paying the price of additional bandwidth.

REFERENCES

1. Nyquist, H., "Certain Topics on Telegraph Transmission Theory," *Trans. Am. Inst. Electr. Eng.*, vol. 47, Apr. 1928, pp. 617–644.
2. Shannon, C. E., "A Mathematical Theory of Communication," *Bell Syst. Tech. J.*, vol. 27, 1948, pp. 379–423, 623–657.
3. Shannon, C. E., "Communication in the Presence of Noise," *Proc. IRE,* vol. 37, no. 1, Jan. 1949, pp. 10–21.
4. Bedrosian, E., "Spectrum Conservation by Efficient Channel Utilization," *Rand Corp., Rep. WN-9275-ARPA,* Contract DAHC-15-73-C-0181, Santa Monica, Calif., Oct. 1975.
5. Odenwalder, J. P., *Error Control Coding Handbook,* Linkabit Corporation, San Diego, Calif., July 15, 1977.
6. Smith, J. G., "Spectrally Efficient Modulation," *Proc. IEEE Int. Conv. Commun. (ICC '77)*, June 1977, pp. 3.1-37–3.1-41.
7. Pasupathy, S., "Minimum Shift Keying: A Spectrally Efficient Modulation," *IEEE Commun. Mag.*, July 1979, pp. 14–22.
8. Gronemeyer, S. A., and McBride, A. L., "MSK and Offset QPSK Modulation," *IEEE Trans. Commun.*, vol. COM-24, Aug. 1976, pp. 809–820.
9. M. K. Simon, "A Generalization of Minimum Shift Keying (MSK) Type Signaling Based upon Input Data Symbol Pulse Shaping," *IEEE Trans. Commun.*, vol. COM24, Aug. 1976, pp. 845–857.
10. Korn, I., *Digital Communications,* Van Nostrand Reinhold Company, Inc., New York, 1985.
11. Oetting, J. D., "A Comparison of Modulation Techniques for Digital Radio," *IEEE Trans. Commun.*, vol. COM27, no. 12, Dec. 1979, pp. 1752–1762.
12. Forney, G. D., Jr. et al., "Efficient Modulation for Bandlimited Channels," *IEEE J. Sel. Areas Commun.*, vol. SAC2, no. 5, Sept. 1984, pp. 632–647.
13. Thomas, C. M., Weidner, M. Y., and Durrani, S. H., "Digital Amplitude-Phase Keying with M-ary Alphabets," *IEEE Trans. Commun.*, vol. COM22, no. 2, Feb. 1974, pp. 168–180.
14. Lucky, R. W., and Hancock, J. C., "On the Optimum Performance of N-ary Systems Having Two Degrees of Freedom," *IRE Trans. Commun. Syst.*, vol. CS10, June 1962, pp. 185–192.
15. Campopiano, C. N., and Glazer, B. G., "A Coherent Digital Amplitude and Phase Modulation Scheme," *IRE Trans. Commun. Syst.*, vol. CS10, June 1962, pp. 90–95.

16. Cahn, C. R., "Combined Digital Phase and Amplitude Modulation Communication Systems," *IRE Trans. Commun. Technol.*, Sept. 1960.
17. Foschini, G. J., Gitlin, R. D., and Weinstein, S. B., "Optimization of Two Dimensional Signal Constellations in the Presence of Gaussian Noise," *IEEE Trans. Commun.*, vol. COM22, no. 1, Jan. 1974, pp. 28–38.
18. Welti, G. R., and Jhong, S. L., "Digital Transmission with Coherent Four-Dimensional Modulation," *IEEE Trans. Inf. Theory*, vol. IT20, no. 4, July 1974, pp. 497–502.
19. Gersho, A., and Lawrence, V. B., "Multidimensional Signal Constellations for Voiceband Data Transmission," *IEEE J. Sel. Areas Commun.*, vol. SAC2, no. 5, Sept. 1984, pp. 687–702.
20. Zetterberg, L. H., and Brandstrom, H., "Codes for Combined Phase and Amplitude Modulated Signals in a Four-Dimensional Space," *IEEE Trans. Commun.*, vol. COM25, no. 9, Sept. 1977, pp. 943–950.
21. Wilson, S. G., Sleeper, H. A., and Srinath, N. K., "Four-Dimensional Modulation and Coding: An Alternative to Frequency Reuse," *IEEE 1984 Intl. Commun. Conf.*, pp. 919–923.
22. Ungerboeck, G., "Channel Coding with Multilevel/Phase Signals," *IEEE Trans. Inf. Theory*, vol. IT28, Jan. 1982, pp. 55–67.
23. Ungerboeck, G., "Trellis-Coded Modulation with Redundant Signal Sets, Part I. Introduction," *IEEE Commun. Mag.*, vol. 25, no. 2, Feb. 1987, pp. 5–11.
24. Ungerboeck, G., "Trellis-Coded Modulation with Redundant Signal Sets, Part II; State of the Art," *IEEE Commun. Mag.*, vol. 25, no. 2, Feb. 1987, pp. 12–21.
25. Thapar, H. K., "Real-Time Application of Trellis Coding to High-Speed Voiceband Data Transmission," *IEEE J. Sel. Areas Commun.*, vol. SAC2, no. 5, Sept. 1984, pp. 648–658.
26. Clark, G. C., Jr., and Cain, J. B., *Error Correction Coding for Digital Communications*, Plenum Press, New York, 1981.
27. Divsalar, D., and Yuen, J. H., "Asymmetric MPSK for Trellis Codes," *GLOBECOM '84*, Nov. 26–29, 1984.
28. Divsalar, D., Simon, M. K., and Yuen, J. H., "Trellis Coding with Asymmetric Modulations," *IEEE Trans. Commun.*, vol. COM35, no. 2, Feb. 1987.

PROBLEMS

7.1. Consider a voice-grade telephone circuit with a bandwidth of 3 kHz. Assume that the circuit can be modeled as an AWGN channel.
(a) What is the capacity of such a circuit if the SNR is 30 dB?
(b) What is the minimum SNR required for a data rate of 4800 bits/s on such a voice-grade circuit?
(c) Repeat part (b) for a data rate of 19,200 bits/s.

7.2. Consider that a 100-kbits/s data stream is to be transmitted on a voice-grade telephone circuit (with a bandwidth of 3 kHz). Is it possible to achieve error-free transmission with a SNR of 10 dB? Justify your answer. If it is not possible, suggest system modifications that might be made.

7.3. Consider a source that produces six messages with probabilities $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$, $\frac{1}{32}$, and $\frac{1}{32}$. Determine the average information content in bits, of a message.

probability of 10^{-5} or better, assume that you are required to choose the modulation, coding, and interleaving from the schemes described below.

8-ary noncoherent FSK

16-ary QAM (matched filter detection)

(127, 92) BCH code, $d_{\min} = 11$

Rate $\frac{1}{2}$, feedback-decoded convolutional code, corrects an average of three symbol errors out of a sequence of 21 symbols

Block interleaver (16 × 32)

Convolutional interleaver (150 × 300)

Choose a modulation technique and if deemed necessary, a coding scheme or a coding/interleaving scheme for the following applications. Justify your choices.

(a) Voice-grade telephone channel with 2400 Hz of usable bandwidth and available $E_b/N_0 = 14$ dB.

(b) Satellite channel with 40 kHz of usable bandwidth and with an available E_b/N_0 of 7.3 dB.

(c) Voice-grade link over a bursty noise channel. A noise burst typically lasts for 100 ms. The usable bandwidth is 3400 Hz and the available $E_b/N_0 = 10$ dB.

7.11. (a) For a fixed error probability, show that the relationship between alphabet size, M, and required average power for MPSK versus QAM can be expressed as

$$\frac{\text{average power for MPSK}}{\text{average power for QAM}} \simeq \frac{3M^2}{2(M-1)\pi^2}$$

(b) Discuss the advantage of one type of signaling over the other.

7.12. Telephone modems operating at 19.2 kbits/s are now available using trellis-coded QAM modulation.

(a) Calculate the bandwidth efficiency of such modems, assuming that the usable channel bandwidth is 2400 Hz.

(b) Assuming AWGN and an available $E_b/N_0 = 10$ dB, calculate the theoretically available capacity in the 2400-Hz bandwidth.

(c) What is the required E_b/N_0 that will enable a 2400-Hz bandwidth to have a capacity of 19.2 kbits/s?

7.13. Figure 7.15 shows several 16-ary symbol constellations.

(a) For the (5, 11) circular constellations, compute the minimum radial distances r_1 and r_2 if the minimum distance between each symbol must be 1 unit.

(b) Compute the average signal power for the (5, 11) circular constellation, and compare it to the average signal power for the 4 × 4 ($M = 16$) square constellation (with the same minimum distance between symbols).

(c) Why might the square constellation be more practical?

7.14. Consider that the rate $\frac{2}{3}$ trellis-coded system of Section 7.10.7 is used over a binary symmetric channel (BSC). Assume that the initial encoder state is the 00 state. At the output of the BSC, the sequence $\mathbf{Z}$ = (1 1 1 0 0 1 1 0 1 0 1 1 rest all "0") is received.

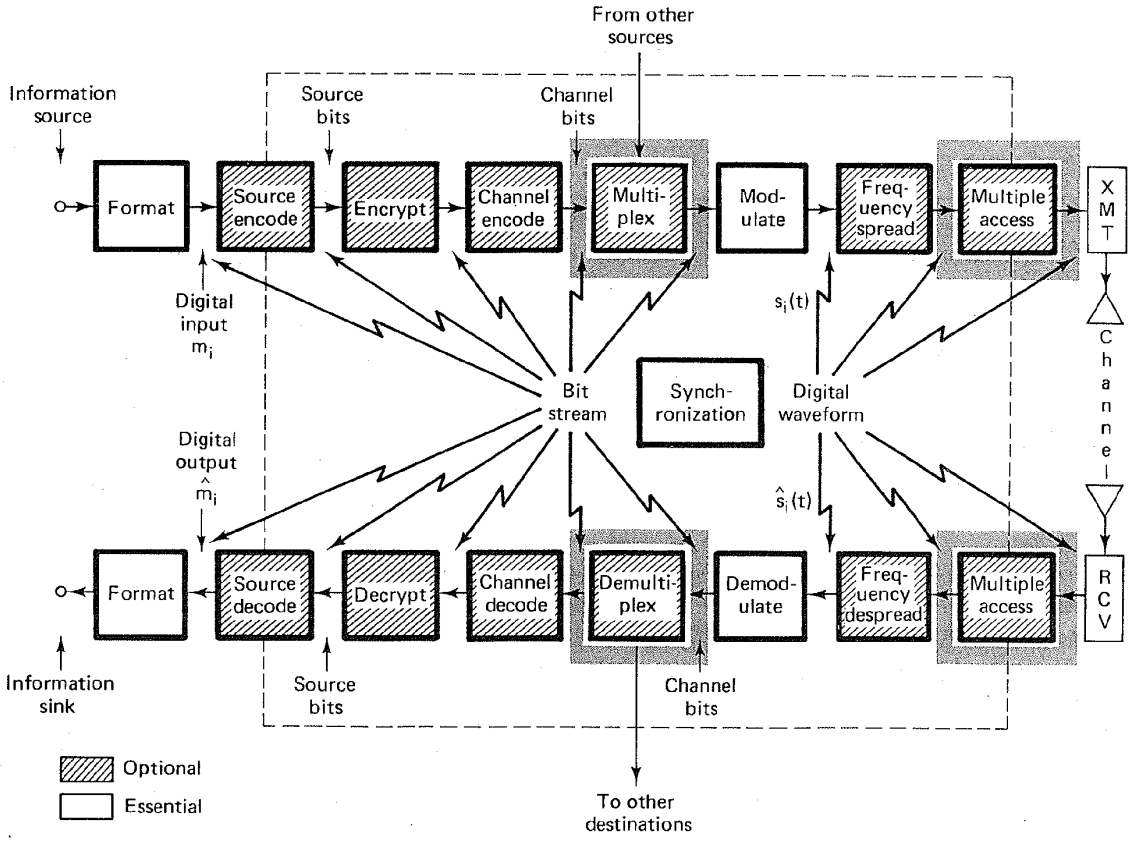
(a) Find the maximum likelihood path through the trellis diagram and determine the first 6 decoded information bits. If a tie occurs between any two merged paths, choose the upper branch entering the particular state.

(b) Determine if any channel bits in $\mathbf{Z}$ had been inverted by the channel during transmission, and if so, identify them.

(c) Explain how you would proceed with the problem if the channel were specified as a Gaussian channel instead of a BSC.

CHAPTER 9

Multiplexing and Multiple Access



The terms "multiplexing" and "multiple access" refer to the sharing of a fixed communications resource (CR). There is a subtle difference between multiplexing and multiple access. With *multiplexing*, users' requirements or plans for CR sharing are fixed, or at most, slowly changing. The resource allocation is assigned a priori, and the sharing is usually a process that takes place within the confines of a *local site* (e.g., a circuit board). *Multiple access*, however, usually involves the *remote sharing* of a resource, such as a satellite. With a dynamically changing multiple access scheme, a system controller must become aware of each user's CR needs; the amount of time required for this information transfer constitutes an overhead and sets an upper limit on the efficiency of the utilization of the CR.

9.1 ALLOCATION OF THE COMMUNICATIONS RESOURCE

There are three basic ways to increase the throughput (total data rate) of a communications resource (CR). The first way is either to increase the transmitter's effective isotropic radiated power (EIRP) or to reduce system losses so that the received E_b/N_0 is increased. The second way is to provide more channel bandwidth. The third approach is to make the allocation of the CR more efficient. This third approach is the domain of communications multiple access. The problem, in the context of a satellite transponder, is to efficiently allocate portions of the transponder's fixed CR to a large number of users who seek to communicate digital information to each other at a variety of bit rates and duty cycles. The

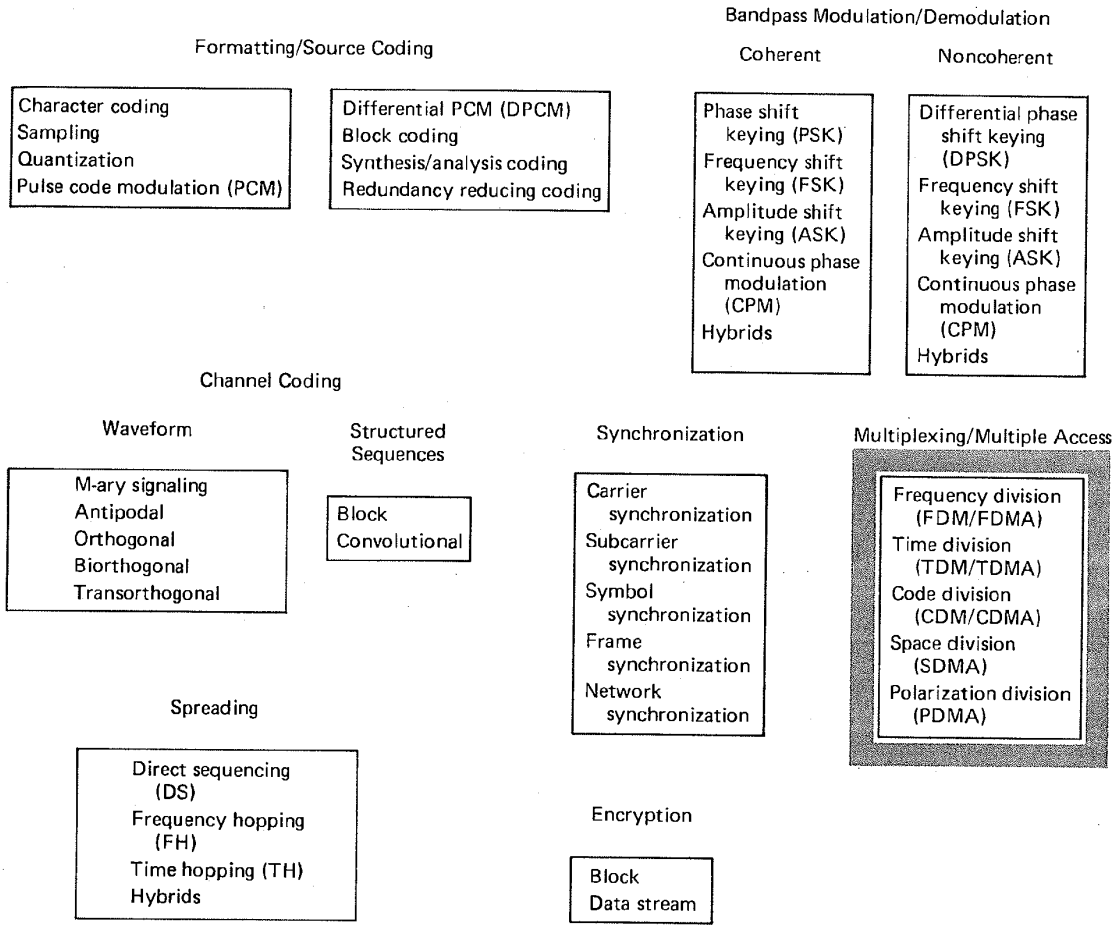


Figure 9.1 Basic digital communication transformations.

basic ways of distributing the communications resource. listed under the heading "multiplexing/multiple access" in Figure 9.1, are:

1. *Frequency division (FD).* Specified subbands of frequency are allocated.
2. *Time division (TD).* Periodically recurring time slots are identified. With some systems, users are provided a fixed assignment in time. With others, users may access the resource at random times.
3. *Code division (CD).* Specified members of a set of orthogonal or nearly orthogonal spread spectrum codes (each using the full channel bandwidth) are allocated.
4. *Space division (SD) or multiple beam frequency reuse.* Spot beam antennas

are used to separate radio signals by pointing in different directions. It allows for reuse of the same frequency band.

5. *Polarization division (PD) or dual polarization frequency reuse.* Orthogonal polarizations are used to separate signals, allowing for reuse of the same frequency band.

The key to *all* multiplexing and multiple access schemes is that various signals share a CR without creating unmanageable interference to each other in the detection process. The allowable limit of such interference is that signals on one CR channel should not significantly increase the probability of error in another channel. Orthogonal signals on separate channels will avoid interference between users. Signal waveforms $x_i(t)$, where $i = 1, 2, \ldots$, are defined to be orthogonal if they can be described in the time domain by

$$\int_{-\infty}^{\infty} x_i(t)x_j(t)\,dt = \begin{cases} K & \text{for } i = j \\ 0 & \text{otherwise} \end{cases} \tag{9.1}$$

where K is a nonzero constant. Similarly, the signals are orthogonal if they can be described in the frequency domain by

$$\int_{-\infty}^{\infty} X_i(f)X_j(f)\,df = \begin{cases} K & \text{for } i = j \\ 0 & \text{otherwise} \end{cases} \tag{9.2}$$

where the functions $X_i(f)$ are the Fourier transforms of the signal waveforms $x_i(t)$. Channelization characterized by orthogonal waveforms, as shown in Equation (9.1), is called time-division multiplexing or time-division multiple access (TDM/TDMA), and that characterized by orthogonal spectra, as shown in Equation (9.2), is called frequency-division multiplexing or frequency-division multiple access (FDM/FDMA).

9.1.1 Frequency-Division Multiplexing/Multiple Access

9.1.1.1 Frequency-Division Multiplex Telephony

In the early days of telephony, a separate pair of wires was needed for each telephone trunk circuit (trunk circuits interconnect intercity switching centers). As illustrated in Figure 9.2, the skies of all the major cities in the world grew dark with overhead wires as the demand for telephone service grew. A major development in the early 1900s, frequency-division multiplex (FDM) telephony, made it possible to transmit several telephone signals simultaneously on a single wire, and thereby transformed the methods of telephone transmission.

The communications resource (CR) is illustrated in Figure 9.3 as the frequency–time plane. The channelized spectrum shown here is an example of FDM or FDMA. The assignment of a signal or user to a frequency band is *long term* or *permanent*; the CR can simultaneously contain several spectrally separate signals. The first frequency band contains signals that operate between frequencies f_0 and f_1, the second between frequencies f_2 and f_3, and so on. The spectral regions between assignments, called *guard bands*, act as buffer zones to reduce

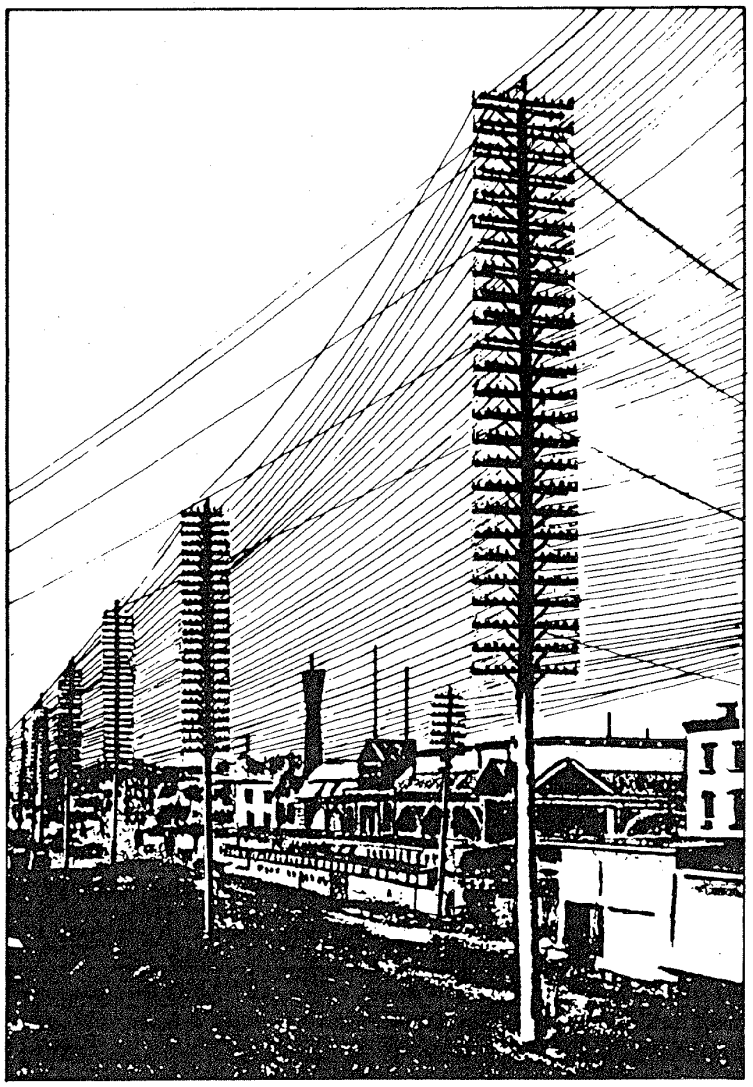


Figure 9.2 In the early days of telephony a pair of wires was needed for each trunk circuit.

interference between adjacent frequency channels. We might ask: How does one transform a baseband signal so that it occupies a higher frequency band? The answer is, by *heterodyning* or *mixing*, also called *modulating* the signal with a fixed frequency from a sine-wave oscillator.

If two input signals to a mixer are sinusoids with frequencies f_A and f_B, the mixing or multiplication will yield new sum and difference frequencies at f_{A+B} and f_{A-B}. The trigonometric identity

$$\cos A \cos B = \tfrac{1}{2}[\cos (A + B) + \cos (A - B)] \tag{9.3}$$

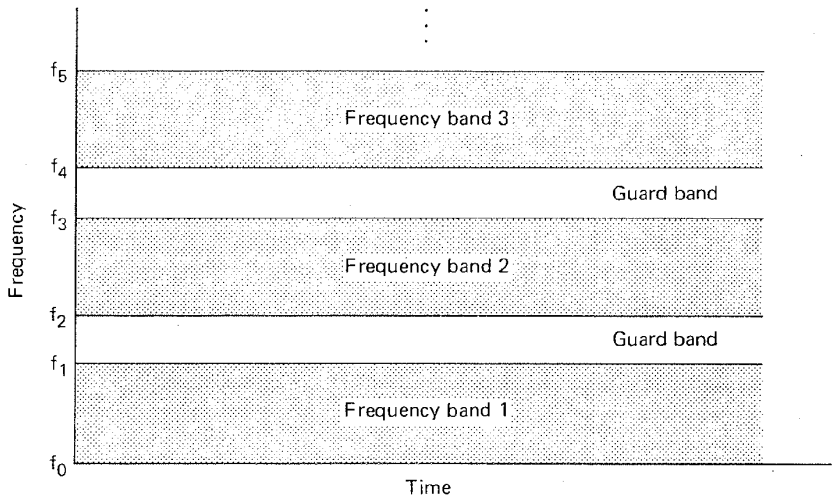


Figure 9.3 Frequency-division multiplexing.

describes the effect of the mixer. Figure 9.4a illustrates the mixing of a typical voice-grade telephone signal, $x(t)$ (baseband frequency range is 300 to 3400 Hz) with a sinusoid from a 20-kHz oscillator. The baseband two-sided magnitude spectrum, $|X(f)|$, is shown in Figure 9.4a. Can the mixer be a linear device? *No.* The output signal of a linear device will only consist of the *same* component frequencies as the input signal, differing only in amplitude and/or phase.

Figure 9.4b illustrates the one-sided magnitude spectrum, $|X(f - f_0)|$, at the mixer output. As a result of the mixing described by Equation (9.3), the output spectrum is a frequency-upshifted version of the baseband spectrum, centered at the oscillator frequency of 20 kHz. This spectrum is called a *double-sideband* (DSB) *spectrum* because the information appears in two different bands of the positive frequency domain. Figure 9.4c shows the lower sideband (LSB), whose frequency range is 16,600 to 19,700 Hz, the result of filtering the DSB spectrum. This sideband is sometimes referred to as the *inverted sideband* because the order of low-to-high frequency components is the reverse of that of the baseband components. Filtering can similarly be used to separate the upper sideband (USB), whose frequency range is 20,300 to 23,400 Hz, as shown in Figure 9.4d. This sideband is sometimes referred to as the *erect sideband* because the order of the low-to-high frequency components corresponds to that of the baseband components. Each sideband of the DSB spectrum contains the same information. Thus, only one sideband, either the USB or the LSB, is needed in order to retrieve the original baseband data.

A simple FDM example with three translated voice channels is seen in Figure 9.5. In channel 1, the 300- to 3400-Hz voice signal is mixed with a 20-kHz oscillator. In channels 2 and 3, a similar type of voice signal is mixed with a 16-kHz and 12-kHz oscillator, respectively. Only the lower sidebands are retained; the

result of the mixing and filtering (to remove the upper sidebands) yields the frequency-shifted voice channels shown in Figure 9.5. The composite output waveform is just the sum of the three signals, having a total bandwidth in the range 8.6 to 19.7 kHz.

Figure 9.6 illustrates the two lowest levels of the FDM multiplex hierarchy for telephone channels. The first level consists of a *group* of 12 channels modulated onto subcarriers shown in the range 60 to 108 kHz. The second level is made up of five groups (60 channels) called a *supergroup* modulated onto the subcarriers shown in the range 312 to 552 kHz. The multiplexed channels are now treated as a composite signal that can be transmitted over cables or can be further modulated onto a carrier wave for radio transmission.

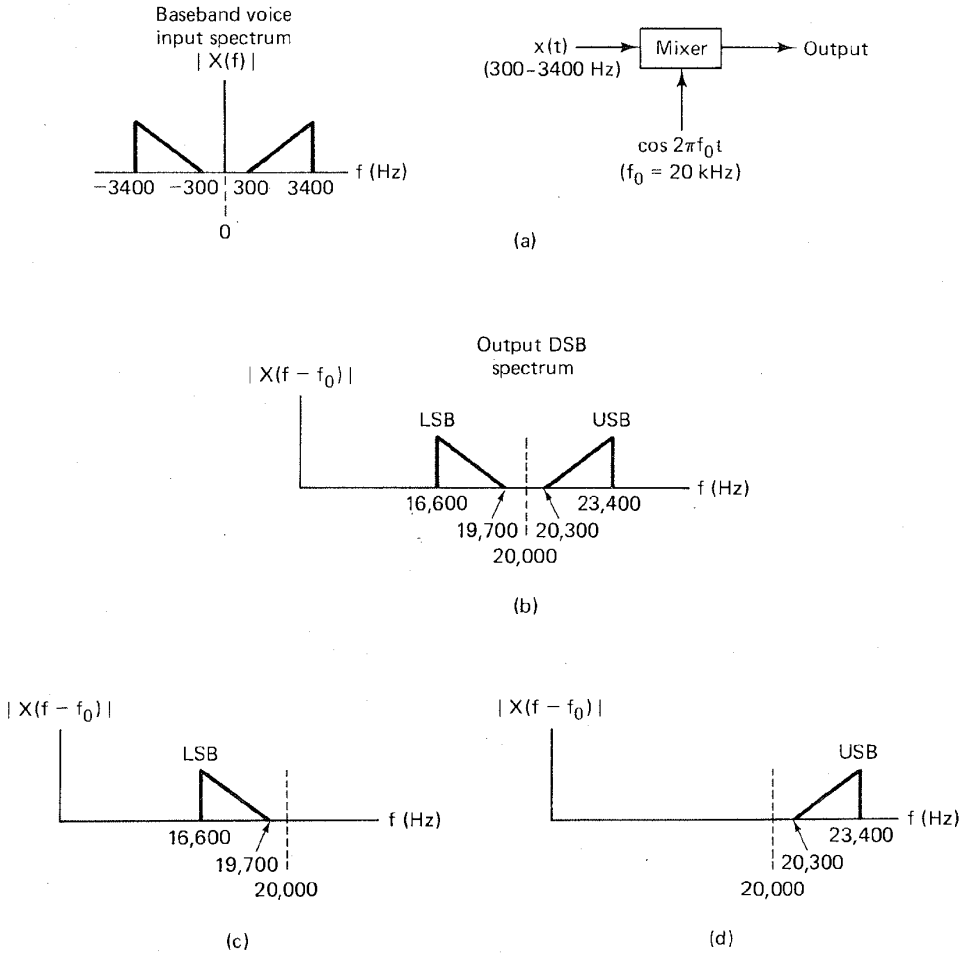


Figure 9.4 Heterodyning (mixing). (a) Mixing operation. (b) Mixer output spectrum. (c) Lower sideband. (d) Upper sideband.

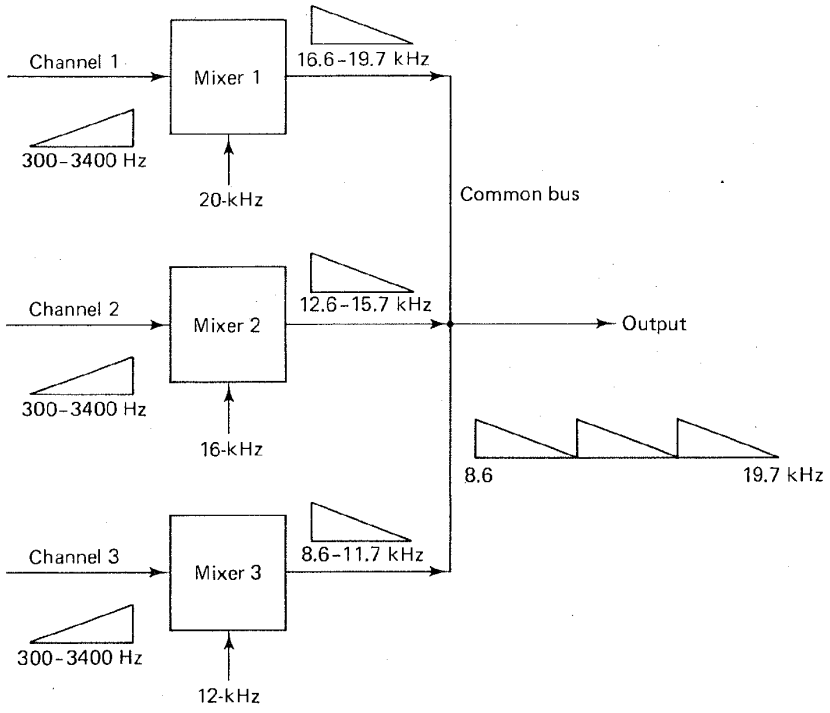


Figure 9.5 Simple FDM example. Three frequency-shifted voice channels.

9.1.1.2 Frequency-Division Multiple Access of Satellite Systems

Most of the free world's communication satellites are positioned in a *geostationary* or *geosynchronous* orbit. This means that the satellite is in a circular orbit, in the same plane as the earth's equatorial plane, and at such an altitude (approximately 19,330 nautical miles) that the orbital period is identical with the earth's rotational period. Since such satellites appear stationary when viewed from the earth, three of them spaced 120° apart can provide worldwide coverage (except for the polar regions). Most communication satellite systems are made up of nonregenerative repeaters or transponders. *Nonregenerative* means that the uplink (earth-to-satellite) transmissions are simply amplified, frequency shifted, and retransmitted on the downlink (satellite-to-earth) without any demodulation/remodulation or signal processing (see Section 4.7.1). The most popular frequency band for commercial satellite communications, called *C-band*, uses a 6-GHz carrier for the uplink and a 4-GHz carrier for the downlink. For C-band satellite systems, *each satellite* is permitted, by international agreement, to use a 500-MHz-wide spectral assignment. Typically, each satellite has 12 transponders with a bandwidth of 36 MHz each. The most common 36-MHz transponders operate in an FDM/FM/FDMA (frequency-division multiplex, frequency-modulated, fre-

quency-division multiple access) multidestination mode. Let us consider each component of this name:

1. *FDM*. Signals such as telephone signals, each one having a single-sideband 4-kHz spectrum (including guard bands) are FDM'd to form a multichannel composite signal.
2. *FM*. The composite signal is frequency-modulated (FM) onto a carrier and transmitted to the satellite.
3. *FDMA*. Subdivisions of the 36-MHz transponder bandwidth may be assigned to different users. Each user receives a specific bandwidth allocation whereby he or she can access the transponder.

Thus, composite FDM channels are FM modulated and transmitted to the satellite within the bandwidth allocation of an FDMA plan. The major advantage of FDMA (compared to TDMA) is its simplicity. The FDMA channels require no

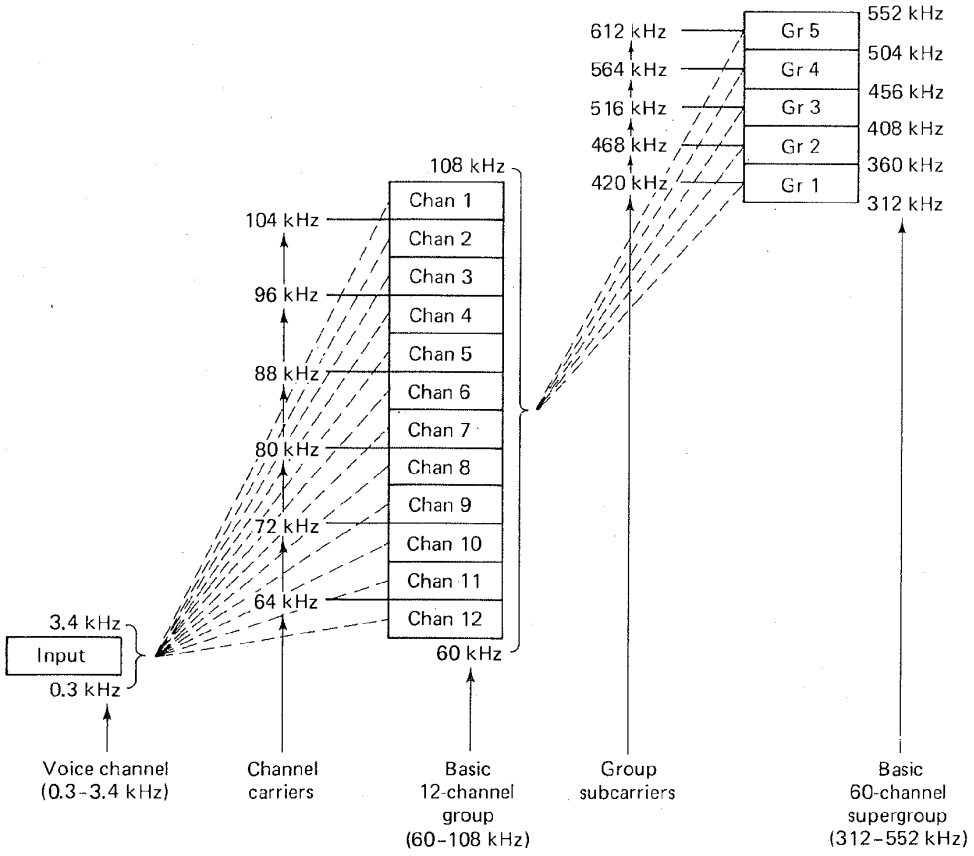


Figure 9.6 Modulation plan of a typical frequency-division multiplex system.

synchronization or central timing; each channel is almost independent of all other channels. Later we discuss some advantages of TDMA compared to FDMA.

9.1.2 Time-Division Multiplexing/Multiple Access

In Figure 9.3, sharing of the communications resource (CR) is accomplished by allocating frequency bands. In Figure 9.7, the same CR is shared by assigning each of M signals or users the full spectral occupancy of the system for a short duration of time called a *time slot*. The unused time regions between slot assignments, called *guard times*, allow for some time uncertainty between signals in adjacent time slots, and thus act as buffer zones to reduce interference. Figure 9.8 is an illustration of a typical TDMA satellite application. Time is segmented into intervals called frames. Each frame is further partitioned into assignable user time slots. The frame structure repeats, so that a fixed TDMA assignment constitutes one or more slots that periodically appear during each frame time. Each earth station transmits its data in bursts, timed so as to arrive at the satellite coincident with its designated time slot(s). When the bursts are received by the satellite transponder, they are retransmitted on the downlink, together with the bursts from other stations. A receiving station detects and demultiplexes the appropriate bursts and feeds the information to the intended user.

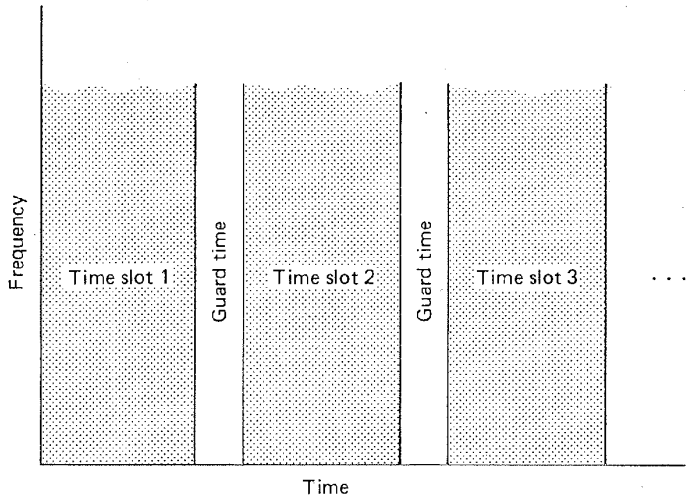


Figure 9.7 Time-division multiplexing.

9.1.2.1 Fixed-Assignment TDM/TDMA

The simplest TDM/TDMA scheme, called *fixed-assignment TDM/TDMA*, is so named because the M time slots that make up each frame are preassigned to signal sources, long term. Figure 9.9 illustrates, in block diagram form, the operation of such a system. The multiplexing operation consists of providing each

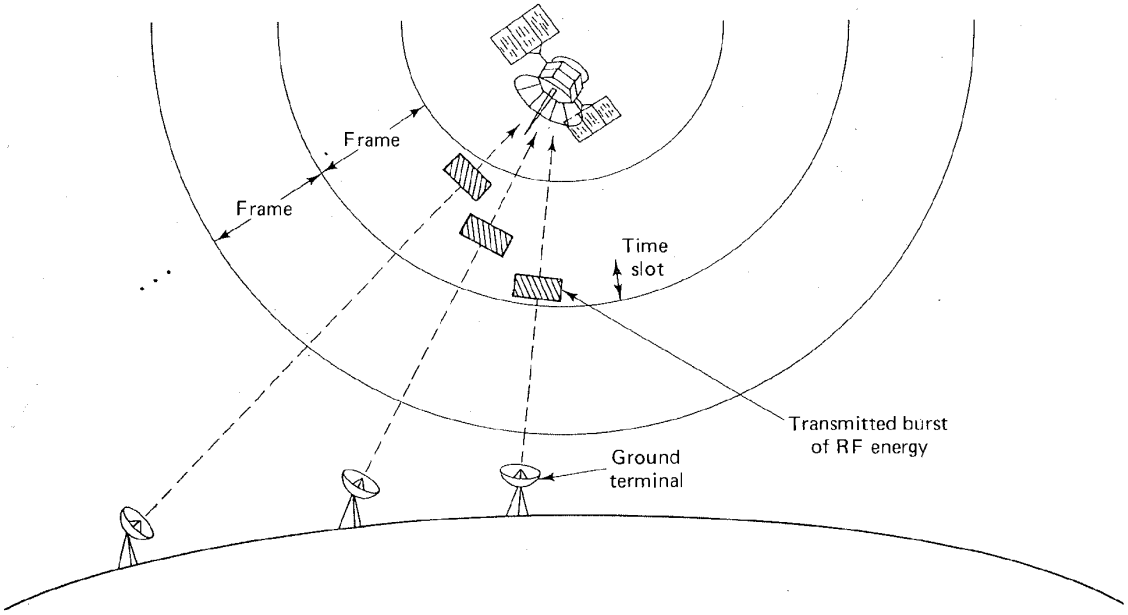


Figure 9.8 Typical TDMA configuration.

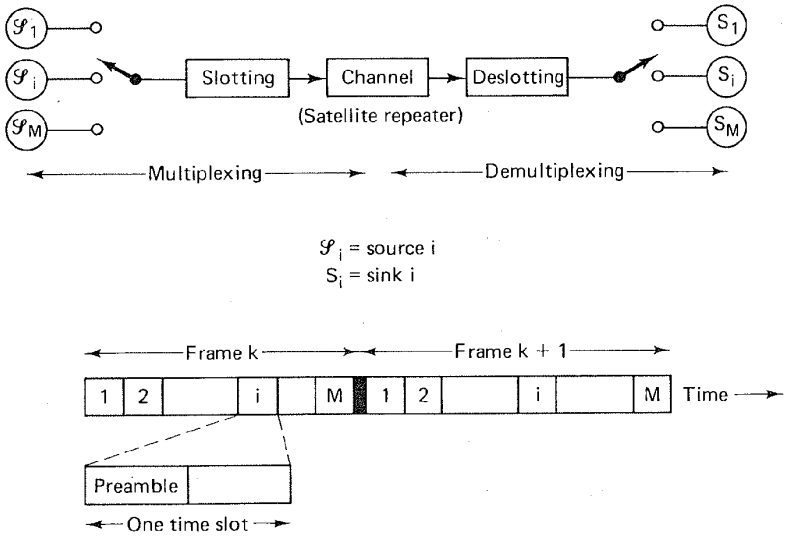


Figure 9.9 Fixed-assignment TDM.

source with an opportunity to occupy one or more slots. The demultiplexing operation consists of deslotting the information and delivering the data to the intended sink. The two commutating switches in Figure 9.9 have to be synchronized so that the message corresponding to source 1, for example, appears on the channel 1 output, and so on. The message itself is generally comprised of a preamble portion and a data portion. The preamble portion usually contains synchronization, addressing, and error control sequences.

A fixed-assignment TDM/TDMA scheme is extremely efficient when the source requirements are predictable, and the traffic is heavy (the time slots are most always filled). However, for bursty or sporadic traffic, the fixed-assignment scheme is wasteful. Consider the simple example shown in Figure 9.10. In this example there are four time slots per frame; each slot is preassigned to users *A*, *B*, *C*, and *D*, respectively. In Figure 9.10a we see a typical activity profile of the four users. During the first frame time, user *C* has no data to transmit; during the second frame time, user *B* has none, and during the third frame time, user *A* has none. In a fixed-assignment TDMA scheme, all of the slots within a frame are preassigned. If the "owner" of a slot has *no* data to send during a particular frame, that slot is wasted. The data stream, shown in Figure 9.10b, illustrates the wasted time slots in this example. When source requirements are unpredictable, as in this example, there can be more efficient schemes, involving the dynamic assignment of the slots rather than a fixed assignment. Such schemes are variously known as packet-switched systems, statistical multiplexers, or concentrators; the effect, shown in Figure 9.10c, is to use all the slots in a frame in such a way that capacity is conserved. In later sections we discuss the TDMA systems used in INTELSAT V and VI.

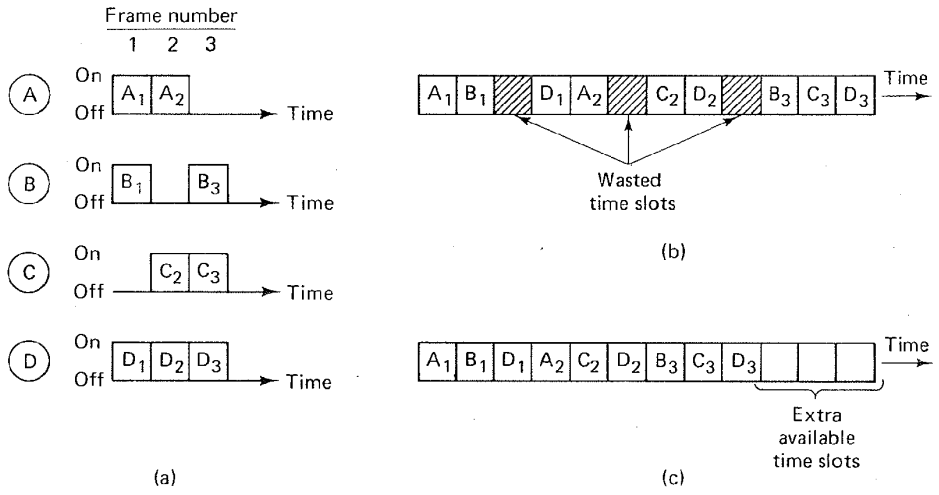


Figure 9.10 Fixed-assignment TDM versus packet switching. (a) Data source activity profiles. (b) Fixed-assignment time-division multiplexing. (c) Time-division packet switching (concentration).

9.1.3 Communications Resource Channelization

In Figure 9.3 we considered that the CR is partitioned into spectral bands, and in Figure 9.7 we viewed the same CR as being partitioned into time slots. Figure 9.11 represents a more general organization of the CR allowing for the assignment of a frequency band for a prescribed period of time. Such a multiple access scheme is referred to as *combined FDMA/TDMA*. For the assignments of frequency bands, let us assume an equal apportionment of the total bandwidth, W, among M user groups or classes, so that M disjoint frequency bands of width W/M hertz are continuously available to their assigned group. Similarly, for the assignment of time slots, the time axis is partitioned into time frames, each of duration T, and the frames are partitioned into N slot times, each of duration T/N. We assume that the users are time synchronized and that the assigned slots are located periodically within the frames. Each user in each frequency band is permitted to transmit during each periodic appearance of the user's assigned slot, and is permitted to use the assigned channel bandwidth for the slot duration. A slot is uniquely determined as the mth slot within the nth frame. Referring to Figure 9.11, we can describe the time of a particular slot (n, m) with reference to time zero as follows:

$$\text{time of slot } (n, m) = nT + \frac{(m-1)T}{N} \le t \le nT + \frac{mT}{N}$$

$$n = 0, 1, \ldots; m = 1, 2, \ldots, N. \quad (9.4)$$

The nth frame time, T, is denoted by the time interval $[nT, (n+1)T]$. As can be seen in Figure 9.11, the domain of the unit signal is the intersection of the time slot (n, m) and the frequency band (j). Assume that a modulation/coding system is chosen so that the full bandwidth W of the CR can support R bits/s. In any

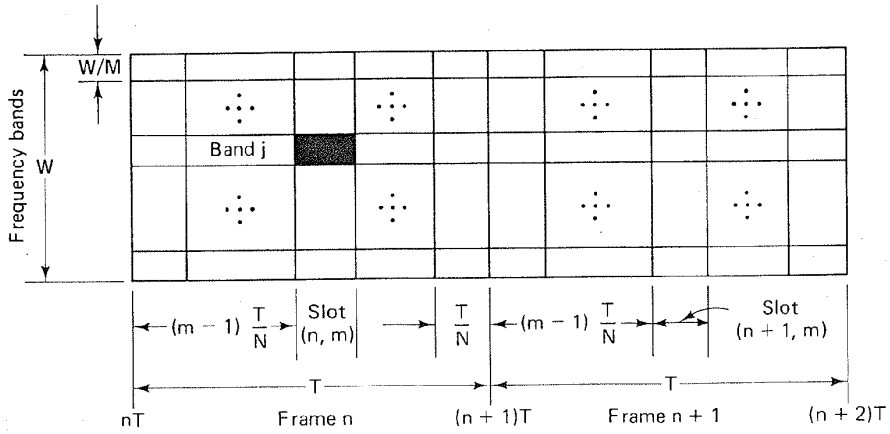


Figure 9.11 Communications resource: time/frequency channelization.

frequency band having a bandwidth of W/M hertz, the associated bit rate will be R/M bits/s. FDMA alone would provide M bands each with a bandwidth of $1/M$ of the full bandwidth of the CR. TDMA alone would provide the full system bandwidth for each of the N slots, where the duration of each slot is $1/N$ of the frame time.

9.1.4 Performance Comparison of FDMA and TDMA

9.1.4.1 Bit Rate Equivalence of FDMA and TDMA

Figure 9.12 highlights the basic differences between an FDMA and TDMA system in a communications resource capable of supporting a total of R bits/s. In Figure 9.12a the system bandwidth is divided into M orthogonal frequency bands. Hence each of the M sources, $\mathcal{S}_m (1 \leq m \leq M)$ can simultaneously transmit at a bit rate of R/M bits/s. In Figure 9.12b the frame is divided into M orthogonal time slots. Hence each of the M sources bursts its transmission at R bits/s, M times faster than the equivalent FDMA user for $(1/M)$th the time. In both cases, the source $\mathcal{S}_m$ transmits information at an average rate of R/M bits/s.

Let the information generated by each of the sources in Figure 9.12 be organized into b-bit groups, or *packets*. In the case of FDMA, the b-bit packets are transmitted in T seconds over each of the M disjoint channels. Therefore, the

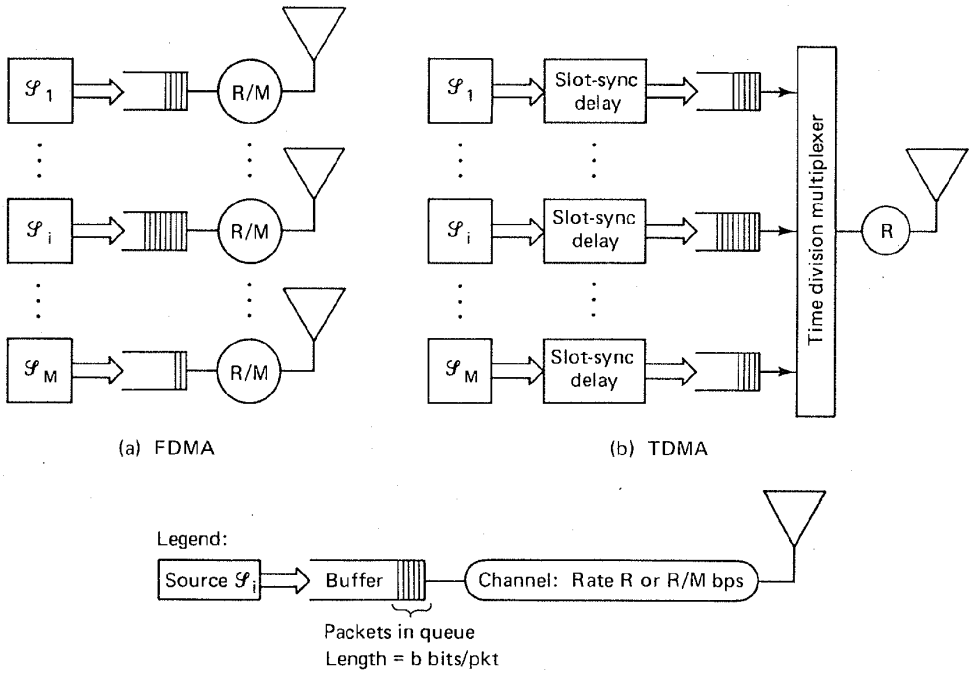


Figure 9.12 (a) FDMA: frequency divided into M orthogonal frequency bands. (b) TDMA: time divided into M orthogonal time slots (one packet per time slot).

total bit rate, R_{FD}, required is

$$R_{FD} = M\frac{b}{T} \quad \text{bits/s} \tag{9.5}$$

In the case of TDMA, the b bits are transmitted in T/M seconds from each source. Therefore, the bit rate, R_{TD}, required is

$$R_{TD} = \frac{b}{T/M} \quad \text{bits/s} \tag{9.6}$$

Since Equations (9.5) and (9.6) yield identical results, we can conclude that

$$R_{FD} = R_{TD} = R = \frac{Mb}{T} \quad \text{bits/s} \tag{9.7}$$

Thus both systems require the same full CR data rate, R bits/s.

9.1.4.2 Message Delays in FDMA and TDMA

From the previous sections it might appear that the duality between FDMA and TDMA will result in equivalent performance. This is not the case when the metric of performance is the average packet *delay*. It can be shown [1, 2] that TDMA is inherently superior to FDMA in the sense that the average packet delay using TDMA is less than the delay using FDMA.

As before, we assume that in the case of FDMA the system bandwidth is divided into M orthogonal frequency bands, and in the case of TDMA the frame is divided into M orthogonal time slots. For the analysis of message delay, the simplest case is that of deterministic data sources. It is assumed that the CR is 100% utilized, so that all frequency bands in the case of FDMA, and all time slots in the case of TDMA, are filled with data packets. For simplicity, it is also assumed that there are *no* overhead costs such as guard bands or guard times.

The message delay, D, can be defined as

$$D = w + \tau \tag{9.8}$$

where w is the average packet waiting time (prior to transmission) and τ is the packet transmission time. In the FDMA case, each packet is sent over a T-second interval, so the packet transmission time for FDMA, τ_{FD}, is simply

$$\tau_{FD} = T \tag{9.9}$$

In the TDMA case, each packet is sent in slots of T/M seconds. We can thus write the TDMA packet transmission time, τ_{TD}, with the use of Equation (9.7), as

$$\tau_{TD} = \frac{T}{M} = \frac{b}{R} \tag{9.10}$$

Since the FDMA channel is continuously available and packets are sent as soon as they are generated, the waiting time, w_{FD}, for FDMA is

$$w_{FD} = 0 \tag{9.11}$$

FDMA and TDMA bit streams are compared in Figure 9.13. For TDMA, Figure 9.13a illustrates that each user's slot begins at a different point in the T-second frame; that is, packet S_{mk} will start at $(m - 1)T/M$ seconds $(1 \leq m \leq M)$ after the packet generation instant. Therefore, the average waiting time, w_{TD}, that a TDMA packet sustains before transmission begins is

$$w_{TD} = \frac{1}{M} \sum_{m=1}^{M} (m - 1) \frac{T}{M} = \frac{T}{M^2} \sum_{n=0}^{M-1} n = \frac{T}{M^2} \frac{(M - 1)(M)}{2}$$

$$= \frac{T}{2}\left(1 - \frac{1}{M}\right) \tag{9.12}$$

The maximum waiting time before transmission of a packet is $(M - 1)T/M$ seconds, and on the average a packet will wait $\frac{1}{2}(M - 1)(T/M) = (T/2)(1 - 1/M)$ seconds, as given by Equation (9.12).

To compare the average delay times, D_{FD} and D_{TD}, for FDMA and TDMA, respectively, we combine Equations (9.9) and (9.11) into Equation (9.8), and sim-

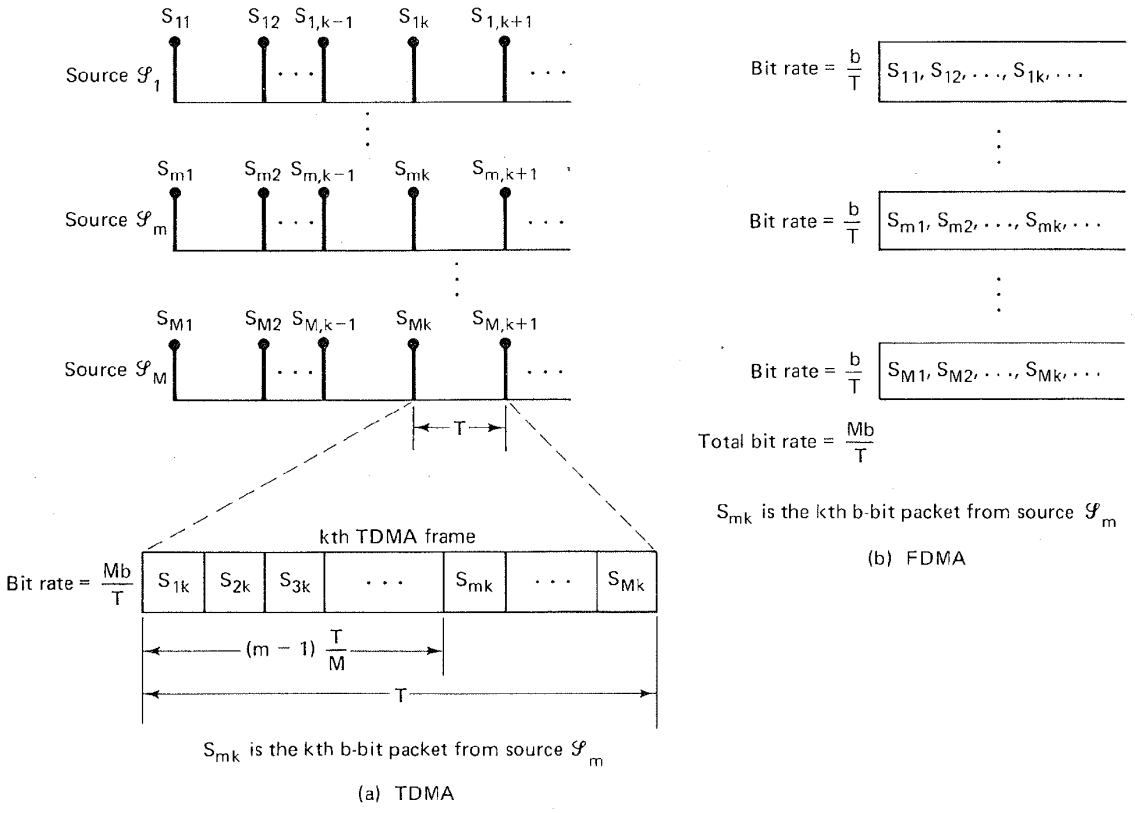


Figure 9.13 (a) TDMA and (b) FDMA channelization.

ilarly combine Equations (9.10) and (9.12) into Equation (9.8), yielding

$$D_{FD} = T \tag{9.13}$$

$$D_{TD} = \frac{T}{2}\left(1 - \frac{1}{M}\right) + \frac{T}{M} = D_{FD} - \frac{T}{2}\left(1 - \frac{1}{M}\right) \tag{9.14}$$

Using Equation (9.7), Equation (9.14) can be written as

$$D_{TD} = D_{FD} - \frac{b}{2R}(M - 1) \tag{9.15}$$

The result indicates that TDMA is inherently superior to FDMA, from a message delay point of view. Although Equation (9.15) assumed that the data source is deterministic, the smaller average message delays for TDMA schemes hold up for any independent message arrival process [1, 2].

9.1.5 Code-Division Multiple Access

In Figure 9.3 the CR plane was illustrated as being shared by slicing it horizontally to form FDMA frequency bands, and in Figure 9.7 the same CR plane was illustrated as being shared by slicing it vertically to form TDMA time slots. These two techniques are the most common choices for multiple access applications. Figure 9.14 illustrates the CR being partitioned by the use of a hybrid combination of FDMA and TDMA known as *code-division multiple access* (CDMA). CDMA

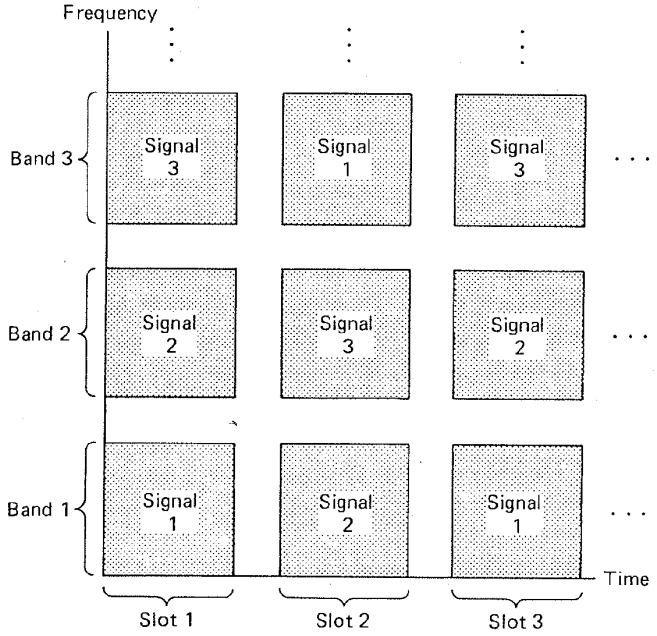


Figure 9.14 Code-division multiplexing.

is an application of spread-spectrum (SS) techniques. Spread-spectrum techniques can be classified into two major categories: *direct-sequence* SS and *frequency hopping* SS. We introduce frequency hopping CDMA (FH-CDMA) in this chapter, and we treat direct-sequence CDMA together with the overall subject of spread-spectrum techniques in Chapter 10.

It is easiest to visualize *frequency hopping* CDMA, illustrated in Figure 9.14, as the short-term assignment of a frequency band to various signal sources. At each successive time slot, whose duration is usually brief, the frequency band assignments are reordered. In Figure 9.14, during time slot 1, signal 1 occupies band 1, signal 2 occupies band 2, and signal 3 occupies band 3. During time slot 2, signal 1 hops to band 3, signal 2 hops to band 1, and signal 3 hops to band 2, and so on. The CR can thus be fully utilized, but the participants, having their frequency bands reassigned at each time slot, appear to be playing "musical chairs." Each user employs a pseudonoise (PN) code, orthogonal (or nearly orthogonal) to all the other user codes, that dictates the frequency hopping band assignments. Details of PN code sequences are treated in Section 10.2. Figure 9.14 is an oversimplified view of the way the CR is shared in frequency hopping CDMA, since the symmetry implies that each frequency hopping signal is in time synchronism with each of the other signals. This is *not the case.* In fact, one of the attractions of CDMA compared to TDMA is that there is no need for synchronization among user groups (only between a transmitter and a receiver within a group).

The block diagram in Figure 9.15 illustrates the frequency hopping modulation process. At each frequency hop time the PN generator feeds a code sequence to a device called a *frequency hopper.* The frequency hopper synthesizes one of the allowable hop frequencies. Assume that the data modulation has an M-ary frequency shift keying (MFSK) format. The essential difference between a conventional MFSK system and a frequency hopping (FH) MFSK system is that in the conventional system, a data symbol modulates a carrier wave that is *fixed* in frequency, but in the hopping system, the data symbol modulates a carrier

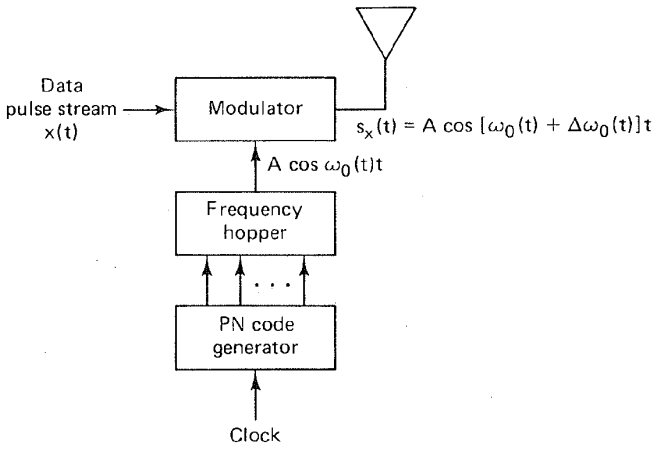


Figure 9.15 CDMA frequency hopping modulation process.

wave that *hops* across the total CR bandwidth. The FH modulation in Figure 9.15 can be thought of as a two-step process—data modulation and frequency hopping modulation—even though it can be implemented in a single step, where the modulator produces a transmission tone based on the simultaneous dictates of the PN code and the data. Frequency hopping systems are covered in detail in Section 10.4.

One might ask: Don't the FDMA and TDMA options provide sufficient multiple access flexibility? FDMA and TDMA methods can surely be relied on to apportion the communications resource equitably. Of what use is this hybrid technique? CDMA offers some unique advantages, as follows:

1. *Privacy*. When the code for a particular user group is only distributed among authorized users, the CDMA process provides communications privacy, since the transmissions cannot easily be intercepted by unauthorized users without the code.
2. *Fading channels*. If a particular portion of the spectrum is characterized by fading, signals in that frequency range are attenuated. In an FDMA scheme, a user who was unfortunate enough to be assigned to the fading position of the spectrum might experience highly degraded communications for as long as the fading persists. However, in a FH-CDMA scheme, only during the time a user hops into the affected portion of the spectrum will the user experience degradation. Therefore, with CDMA, such degradation is shared among all the users.
3. *Jam resistance*. During a given CDMA hop, the signal bandwidth is identical to the bandwidth of conventional MFSK, which is typically equal to the minimum bandwidth necessary to transmit the MFSK symbol. However, over a duration of many time slots, the system will hop over a frequency band which is much wider than the data bandwidth. We refer to this utilization of bandwidth as spread spectrum. In Chapter 10 we develop, in detail, the resistance to jamming that spread spectrum affords a user.
4. *Flexibility*. The most important advantage of CDMA schemes, compared to TDMA, is that there need be no precise time coordination among the various simultaneous transmitters. The orthogonality between user transmissions on different codes is not affected by transmission-time variations. This will become clear upon closer examination of the autocorrelation and cross-correlation properties of the codes, considered in Chapter 10.

9.1.6 Space-Division and Polarization-Division Multiple Access

Figure 9.16a depicts the INTELSAT IVA application of space-division multiple access (SDMA), also called *multiple-beam frequency reuse*. INTELSAT IVA used a dual-beam receive antenna feeding two receivers to allow simultaneous access of the satellite from two different regions of the earth. The frequency band

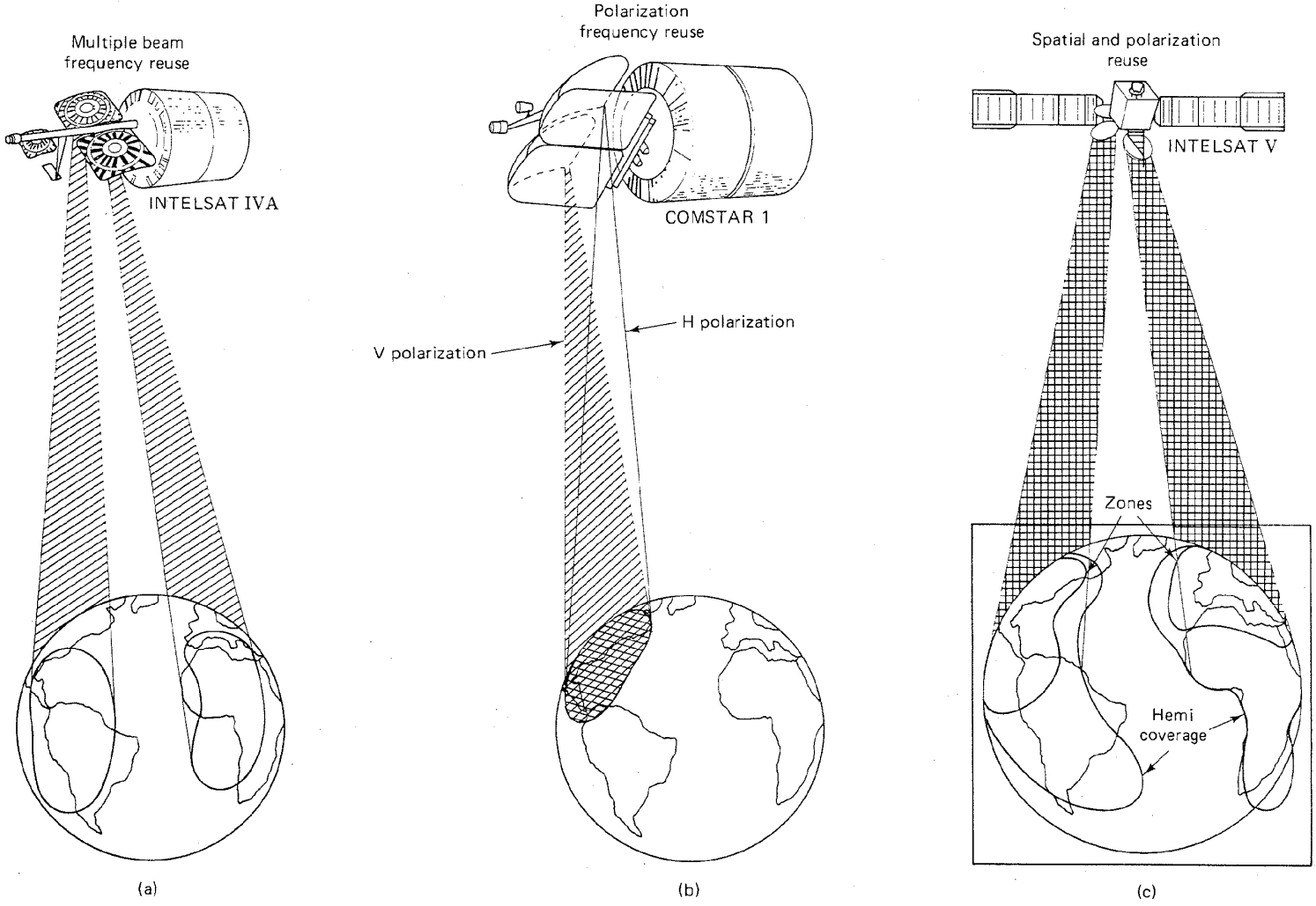


Figure 9.16 SDMA and PDMA. (a) INTELSAT IVA. (b) COMSTAR 1. (c) INTELSAT V (Atlantic coverage).

allocated to each receive beam was identical because the uplink signals were spatially separated. In such cases, the frequency band is said to be *reused.*

Figure 9.16b depicts an application of polarization-division multiple access (PDMA), also called *dual-polarization frequency reuse*, from COMSTAR 1. Here separate antennas are used, each with different polarization and followed by separate receivers, allowing simultaneous access of the satellite from the same region of the earth. Each corresponding earth station antenna needs to be polarized in the same way as its counterpart in the satellite. (This is generally accomplished by providing each participating earth station with an antenna that has dual polarization.) The frequency band allocated to each antenna beam can be identical because the uplink signals are orthogonal in polarization. As with SDMA, the frequency band in PDMA is said to be reused. Figure 9.16c depicts an application of the simultaneous use of SDMA and PDMA in INTELSAT V. There are two separate hemispheric coverages, west and east. There are also two smaller zone beams; each zone beam overlaps a portion of one of the hemispheric beams and is separated from it by orthogonal polarization. Thus there is a fourfold reuse of the spectrum.

9.2 MULTIPLE ACCESS COMMUNICATIONS SYSTEM AND ARCHITECTURE

A *multiple access protocol* or *multiple access algorithm* (MAA) is that rule by which a user knows how to use time, frequency, and code functions to communicate through a satellite to other users. A multiple access system is a com-

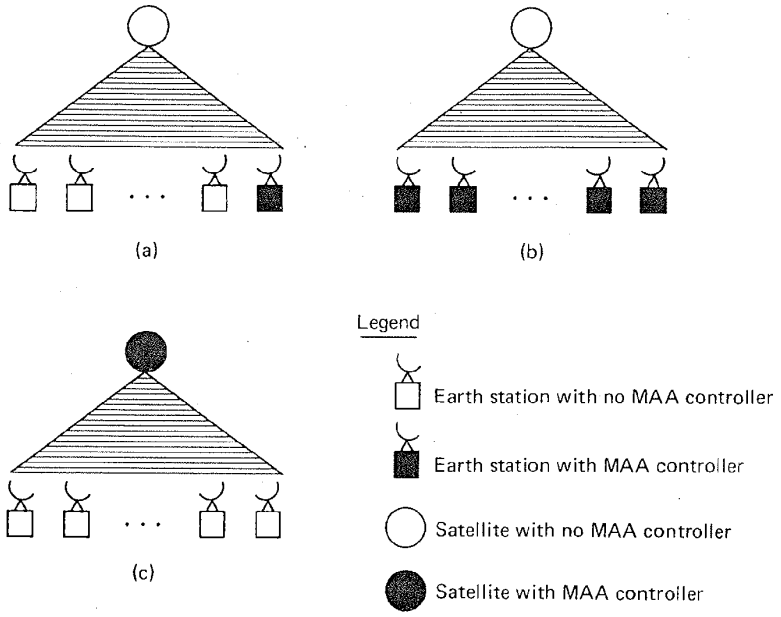


Figure 9.17 Satellite multiple access architecture. (a) Single earth station control. (b) Distributed earth station control. (c) Satellite control.

bination of hardware and software that supports the MAA. The general goal of a multiple access system is to provide communications service in a timely, orderly, and efficient way.

Figure 9.17 illustrates some basic choices for the architecture of a satellite multiple access system. The legend indicates the symbols used for an earth station with and without an MAA controller, and a satellite with and without an MAA controller. Figure 9.17a illustrates the case where one earth station is designated as the master, or the controller. This earth station possesses an MAA computer and responds to the service requests of all other users. Notice that a user's request entails a transmission through the satellite and back down to the controller. The controller's response entails another transmission through the satellite; hence there are two up- and downlink transmissions required for each service assignment. Figure 9.17b illustrates the case where the MAA control is distributed among all the earth stations; there is no single controller. Each earth station uses the same algorithm and they each have identical knowledge regarding access requests and assignments; therefore, only one round trip is required for each service assignment. Figure 9.17c illlustrates the case where the MAA controller is in the satellite. A service request goes from user to satellite, and the response from the satellite can follow immediately; therefore, only one round trip is required for each service assignment.

9.2.1 Multiple Access Information Flow

Figure 9.18 is a flow diagram describing the basic flow of information between the multiple access algorithm (MAA) or controller and an earth station; the numbers below correspond to those on the figure. Recall from the preceding section that the control may be lodged in the satellite, in a master station, or distributed among all the earth stations.

1. *Channelization*. This term refers to the most general allocation information [e.g., channels 1 to N may be allocated for the Army and channels $(N + 1)$

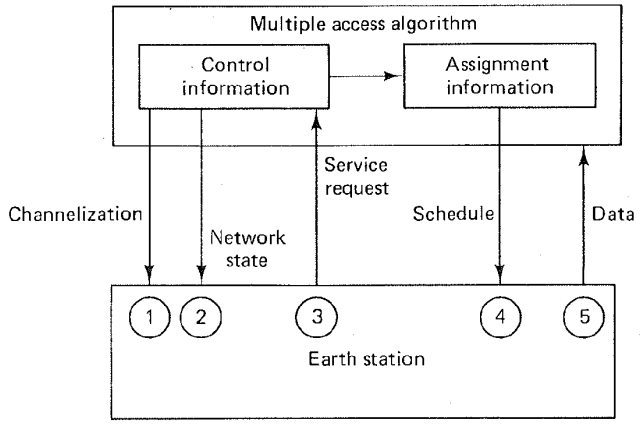


Figure 9.18 Multiple access information flow.

to *M* for the Navy]. This information seldom changes, and may be distributed to the earth stations by the use of a newsletter rather than via the communication system.

2. *Network state* (*NS*). This term refers to the state of the CR. A station is advised regarding the availability of the communications resource and where in the resource (e.g., time, frequency, code position) to transmit its service request(s).
3. *Service request*. Then the station makes its request(s) for service (e.g., allocation for *m* message slots).
4. Upon receipt of the service request(s), the controller sends the station a schedule regarding where and when to position its data in the CR.
5. The station transmits its data according to its assigned schedule.

9.2.2 Demand-Assignment Multiple Access

Multiple access schemes are termed *fixed assignment* when a station has periodic access to the channel independent of its actual need. By comparison, dynamic assignment schemes, sometimes called *demand-assignment multiple access* (DAMA), give the station access to the channel only when it requests access. If the traffic from a station tends to be burst-like or intermittent, DAMA procedures can be much more efficient than fixed-assignment procedures. A DAMA scheme capitalizes on the fact that actual demand *rarely* equals the peak demand. If a system's capacity is equal to the total peak demand and if the traffic is bursty, the system will be underutilized most of the time. However, by using buffers and DAMA, a system with reduced average capacity can handle bursty traffic, at the cost of some queueing delay. Figure 9.19 summarizes the difference between a fixed system, whose capacity is equal to the sum of the user requirements, and a dynamic system, whose capacity is equal to the average of the user requirements.

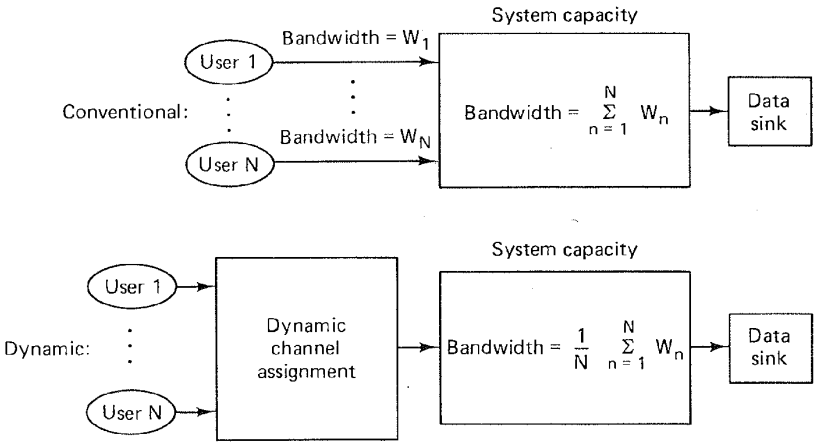


Figure 9.19 Bandwidth reduction for systems using dynamic channel assignment.

9.3 ACCESS ALGORITHMS

9.3.1 ALOHA

In 1971, the University of Hawaii began operation of its ALOHA system. A communication satellite was used to interconnect the several university computers by use of a random access protocol [3–7]. The system concept was extremely simple, consisting of the following modes:

1. *Transmission mode.* Users transmit at any time they desire, encoding their transmissions with an error detection code.
2. *Listening mode.* After a message transmission, a user listens for an acknowledgment (ACK) from the receiver. Transmissions from different users will sometimes overlap in time, causing reception errors in the data in each of the contending messages. We say that the messages have *collided.* In such cases, the errors are detected, and the users receive a negative acknowledgment (NAK).
3. *Retransmission mode.* When a NAK is received, the messages are simply retransmitted. Of course, if the colloding users were to retransmit immediately, they would collide again. Therefore, the users retransmit after a *random* delay.
4. *Timeout mode.* If, after a transmission, the user does not receive either an ACK or NAK within a specified time, the user retransmits the message.

9.3.1.1 Message Arrival Statistics

Assume that the total system demand requires an average message or packet arrival rate of λ successful or accepted messages per second. Because of the presence of collisions, some of the messages will be unsuccessful or rejected. Therefore, we define the total traffic arrival rate, λ_t, as the acceptance rate, λ, plus the rejection rate, λ_r, as follows:

$$\lambda_t = \lambda + \lambda_r \tag{9.16}$$

Let us denote the length of each message or packet as b bits. Then we can define the average amount of successful traffic or *throughput*, ρ', on the channel in units of bits per second, as

$$\rho' = b\lambda \tag{9.17}$$

We can also define the *total traffic*, G', on the channel in units of bits per second, as

$$G' = b\lambda_t \tag{9.18}$$

With the channel capacity (maximum bit rate) designated as R bits per second, let us further define a *normalized throughput*, ρ, and a *normalized total traffic*, G, as

$$\rho = \frac{b\lambda}{R} \tag{9.19}$$

$$G = \frac{b\lambda_t}{R} \tag{9.20}$$

Normalized throughput, ρ, expresses throughput as a fraction ($0 \leq \rho \leq 1$) of channel capacity. Normalized total traffic, G, expresses total traffic as a fraction ($0 \leq G \leq \infty$) of the channel capacity. Notice that G can take on values greater than unity.

We can also define the transmission time of each packet as folllows:

$$\tau = \frac{b}{R} \quad \text{seconds/packet} \tag{9.21}$$

By substituting equation (9.21) into Equations (9.19) and (9.20), we can write

$$\rho = \lambda\tau \tag{9.22}$$

$$G = \lambda_t\tau \tag{9.23}$$

A user can successfully transmit a message as long as no other user began one within the previous τ seconds or starts one within the next τ seconds. If another user began a message within the previous τ seconds, its tail end will collide with the current message. If another user begins a message within the next τ seconds, it will collide with the tail end of the current message. Thus a space of 2τ seconds is needed for each message.

The message arrival statistics for unrelated users of a communication system is often modeled as a Poisson process. The probability of having K new messages arrive during a time interval of τ seconds is given by the Poisson distribution [8] to be

$$P(K) = \frac{(\lambda\tau)^K e^{-\lambda\tau}}{K!} \qquad K \geq 0 \tag{9.24}$$

where λ is the average message arrival rate. Because the users transmit without regard for each other in the ALOHA system, this expression is useful for calculating the probability that exactly $K = 0$ other messages are transmitted during a time interval 2τ. This is the probability, P_s, that a user's message transmission was successful (experienced no collisions). To compute P_s, assuming that all traffic is Poisson, we use λ_t and 2τ in Equation (9.24). Thus

$$P_s = P(K = 0) = \frac{(2\tau\lambda_t)^0 e^{-2\tau\lambda_t}}{0!} = e^{-2\tau\lambda_t} \tag{9.25}$$

In Equation (9.16) we defined total traffic arrival rate λ_t, in terms of the successful portion, λ, and the repetition or unsuccessful portion, λ_r; then, by definition, the probability of a successful packet can be expressed as

$$P_s = \frac{\lambda}{\lambda_t} \tag{9.26}$$

By combining Equations (9.25) and (9.26), we have

$$\lambda = \lambda_t e^{-2\tau\lambda_t} \tag{9.27}$$

By combining Equation (9.27) with Equations (9.22) and (9.23), we can write

$$\rho = Ge^{-2G} \tag{9.28}$$

Equation (9.28) relates the normalized throughput, ρ, to the normalized total traffic, G, on the channel for the ALOHA system. A plot of this relationship labeled "pure ALOHA" is shown in Figure 9.20. As G increases, ρ increases until a point is reached where further traffic increases create a large enough collision rate to cause a reduction in the throughput. The maximum ρ, equal to $1/2e = 0.18$, occurs at a value of $G = 0.5$. Therefore, for a pure ALOHA channel, only 18% of the CR can be utilized. Simplicity of control is achieved at the expense of channel capacity [7, 9].

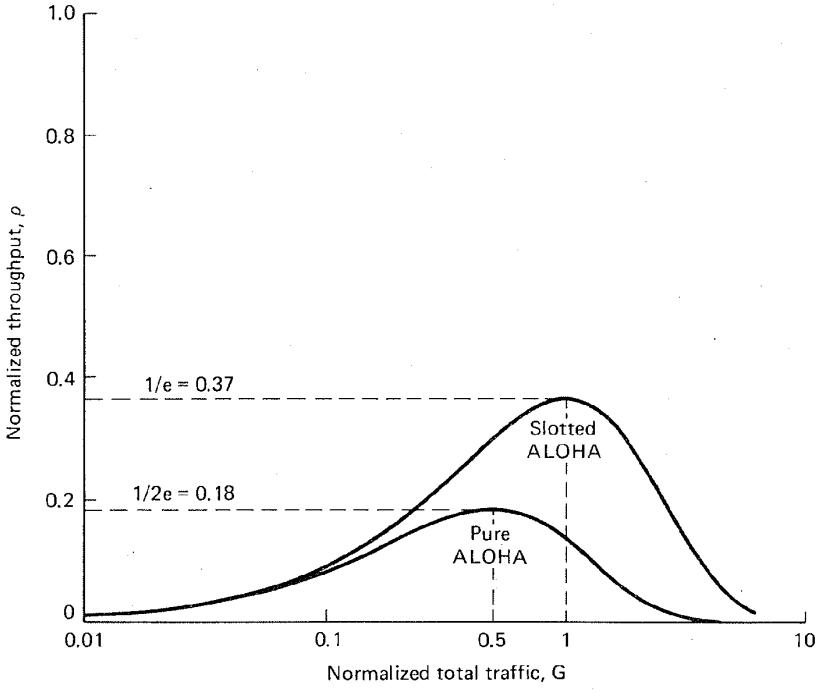


Figure 9.20 Throughput in ALOHA channels (successful transmissions versus total transmissions).

9.3.2 Slotted ALOHA

The pure ALOHA scheme can be improved by requiring a small amount of coordination among the stations. The slotted ALOHA (S-ALOHA) is such a system. A sequence of synchronization pulses is broadcast to all stations. As with pure

ALOHA, packet lengths are constant. Messages are required to be sent in the slot time between synchronization pulses, and can be started only at the *beginning* of a time slot. This simple change reduces the rate of collisions by half, since only messages transmitted in the same slot can interfere with one another. It can be shown [9, 10] that for S-ALOHA, the reduction in the *collision window* from 2τ to τ results in the following relationship between normalized throughput, ρ, and normalized total traffic, G.

$$\rho = Ge^{-G} \tag{9.29}$$

The plot of Equation (9.29) is shown in Figure 9.20 labeled "slotted ALOHA." Here the maximum value of ρ is $1/e = 0.37$, or an improvement of two times the pure ALOHA protocol.

The retransmission mode described for the pure ALOHA system was modified for S-ALOHA so that if a negative acknowledgment (NAK) occurs, the user retransmits after a *random* delay of an integer number of slot times. Figure 9.21 illustrates the S-ALOHA operation. A packet of data bits is shown transmitted by user k followed by the satellite acknowledgment (ACK). Also shown are users m and n simultaneously transmitting packets, which results in a collision; a NAK is returned. Each using station employs a random-number generator to select its retransmission time. The figure illustrates an example of the m and n retransmission at their respective randomly selected times. Of course, there is some probability that users m and n will recollide. However, in that case, they simply repeat the retransmission, using another random delay.

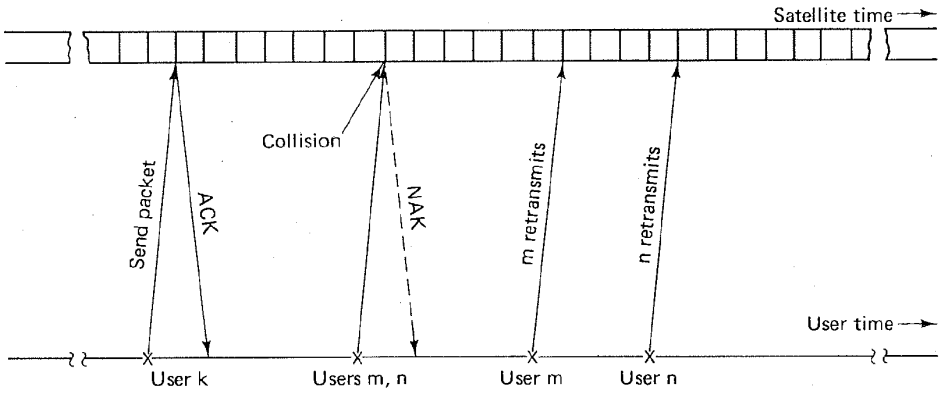


Figure 9.21 Random access scheme: slotted ALOHA operation.

Example 9.1 Poisson Process

Assuming that packet transmissions and retransmission can both be described as a Poisson process, calculate the *probability* that a data packet transmission in an S-ALOHA system will experience a collision with *one other user*. Assume that the total traffic rate $\lambda_t = 10$ packets/s and the packet duration $\tau = 10$ ms.

Solution

$$P(K = 1) = \frac{(\tau\lambda_t)^K e^{-\tau\lambda_t}}{K!}\bigg|_{K=1}$$

$$= (10 \times 0.01)^1 e^{-0.1} = 0.1e^{-0.1}$$

$$= 0.09$$

9.3.3 Reservation-ALOHA

A significant improvement was made to the ALOHA system with the introduction of the reservation-ALOHA (R-ALOHA) [11] scheme. The R-ALOHA system has two basic modes: an unreserved mode and a reserved mode; each is described below.

Unreserved Mode (Quiescent State)

1. A time frame is established and divided into a number of small reservation subslots.
2. Users use these small subslots to reserve message slots.
3. After requesting a reservation, the user listens for an acknowledgment and a slot assignment.

Reserved Mode

1. The time frame is divided into $M + 1$ slots whenever a reservation is made.
2. The first M slots are used for message transmissions.
3. The last slot is subdivided into subslots to be used for reservation/requests.
4. Users send message packets only in their assigned portions of the M slots.

Consider the R-ALOHA example shown in Figure 9.22. In the quiescent state, with no reservations, time is partitioned into short subslots for making reservations. Once a reservation is made, the system is configured so that $M = 5$ message slots followed by $V = 6$ reservation subslots becomes the timing format. The figure illustrates a request and an acknowledgment in progress. In this example the station seeks to reserve three message slots. The reservation acknowledgment advises the using station where to locate its first data packet. Since the control is distributed so that all participants receive the downlink transmissions and are thus aware of the reservations and time format, the acknowledgment need not disclose any more than the location of the first slot. As shown in Figure 9.22, the station sends its second packet in the slot following the first packet. The user further knows that the next slot is comprised of six subslots for reservations, so *no* packets are transmitted during this time. The third and final packet is sent in the following slot. When there are no reservations taking place, the system reverts back to its quiescent format of subslots only. Since the control is distributed, all the participants are made aware of the quiescent format by receiving appropriate

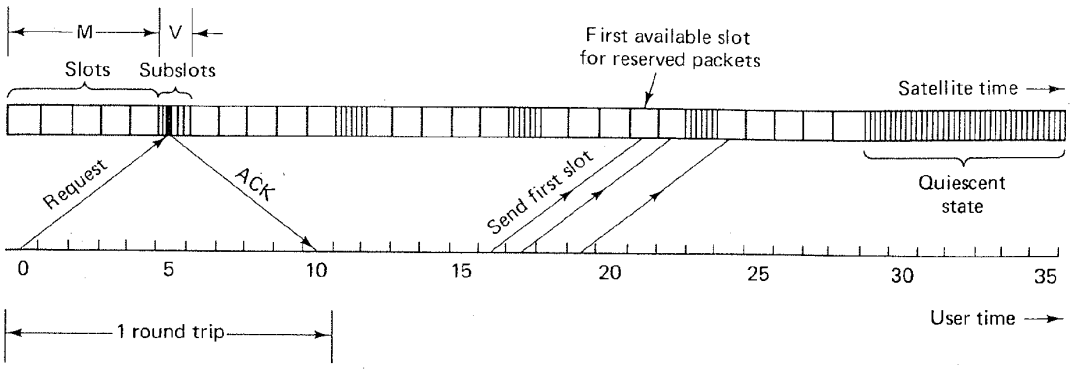


Figure 9.22 Example of reservation ALOHA. Station seeks to reserve three slots (M = 5 slots, V = 6 subslots).

synchronizing pulses on the downlink. Other interesting reservation schemes are discussed in References [12, 13].

9.3.4 Performance Comparison of S-ALOHA and R-ALOHA

From Chapter 3 the basic quality measure of a digital modulation scheme is its P_B versus E_b/N_0 curve. This measure is particularly useful because E_b/N_0 is a *normalized signal-to-noise ratio*; being normalized, the curves allow us to compare the performance of various modulation schemes. There is a similar performance measure for multiple access schemes. Here we are interested in the average delay versus normalized throughput. What would an *ideal delay–throughput curve* look like? Figure 9.23 illustrates such a curve. For normalized throughput values

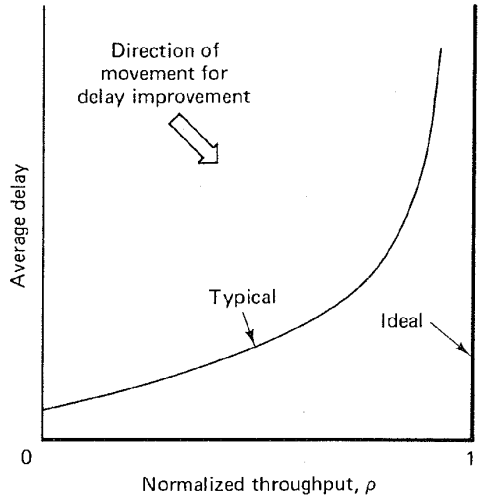


Figure 9.23 Delay–throughput characteristic.

of, $0 \leq \rho < 1$, the delay equals zero until $\rho = 1$; then the delay increases without bound. Figure 9.23 also shows a *typical* delay–throughput curve and the direction in which the curve will move as delay performance improves.

Figure 9.24 compares the delay–throughput performance of S-ALOHA with that of R-ALOHA (formatted with two message slots and six reservation subslots). Knowing the location of the *ideal* curve it is easy to compare the delay performance of these two systems. For a throughput of less than approximately 0.20, the S-ALOHA manifests less average delay than does R-ALOHA. But for values of ρ between 0.20 and 0.67, it is apparent that R-ALOHA is superior, since the average delay is less. Why does the S-ALOHA perform better at low traffic intensity? The S-ALOHA algorithm does not require the overhead of the reservation subslots as does R-ALOHA. Therefore, at low values of ρ, R-ALOHA pays the price of greater delay due to the greater overhead. For $\rho > 0.2$, the collisions and retransmissions inherent in the S-ALOHA system cause it to incur greater delay (unbounded at $\rho = 0.37$), more quickly than the R-ALOHA system. At higher throughput ($0.2 < \rho < 0.67$), the overhead structure of R-ALOHA ensures that its delay degradation grows in a more orderly manner than S-ALOHA. For R-ALOHA, an unbounded delay is not reached until $\rho = 0.67$.

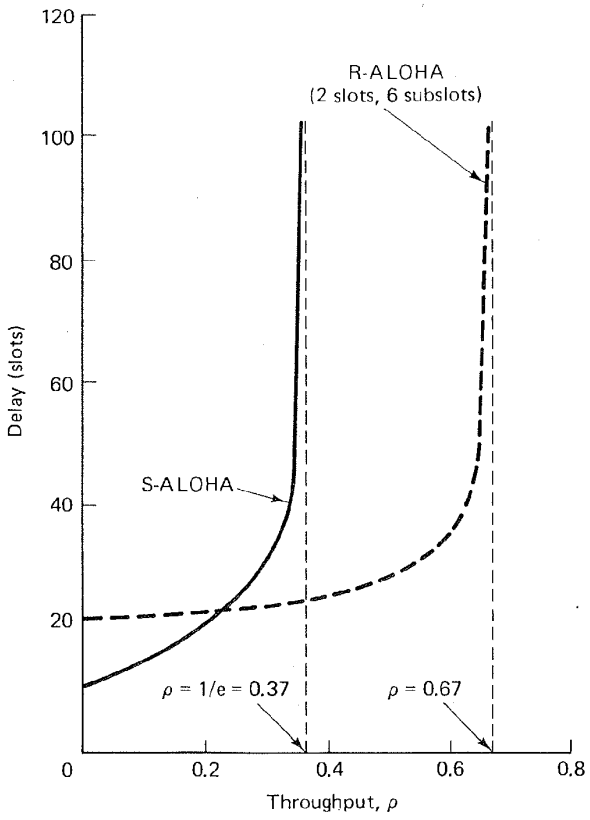


Figure 9.24 Delay–throughput comparison: S-ALOHA versus R-ALOHA on a satellite channel.

Example 9.2 Channel Utilization

(a) Normalized throughput, ρ, is a measure of channel utilization. It can be found by forming the ratio of the successfully transmitted message traffic, in bits per second to the total message traffic, including rejected messages, in bits per second. Calculate the normalized throughput of a channel that has a maximum data rate $R = 50$ kbits/s and operates with $M = 10$ ground stations, each station transmitting at the average rate of $\lambda = 2$ packets/second. The system format provides for $b = 1350$ bits/packet.

(b) Which of the three ALOHA schemes discussed—pure, slotted, and reservation—could be successfully used with this channel?

Solution

(a) Generalizing Equation (9.19) to allow for traffic from multiple stations, we have

$$\rho = \frac{Mb\lambda}{R}$$

$$= \frac{10(1350)(2)}{50{,}000}$$

$$= 0.54$$

(b) Only the R-ALOHA scheme could be used for this system, since with each of the other schemes, 54% of the resource cannot be utilized.

9.3.5 Polling Techniques

One way to impose order on a system with multiple users having random access requirements is to institute a controller that periodically polls the user population to determine their service requests. If the user population is large (e.g., thousands of terminals) and the traffic is bursty, the time required to poll the population can be an excessive overhead burden. One technique for rapidly polling a user population [4, 14] is called a *binary tree search*. Figure 9.25 illustrates a satellite example of such a tree search to resolve contention among users. In this example, assume that the total user population is eight terminals; let them be identified by the binary numbers 000 to 111 as shown in Figure 9.25. Assume that terminals 001, 100, and 110 are contending for the service of a single channel. The tree search operates by continually partitioning the population until there is just a single branch remaining. The terminal corresponding to that branch is the "winner" and hence the first terminal to access the channel. The operation is repeated and again yields a single terminal that may next use the channel. The algorithm proceeds according to the following steps (see Figure 9.25):

1. The satellite requests the transmission of the contending terminals' first (leftmost) bit of their identification (ID) numbers.
2. Terminal 001 transmits a zero, and terminals 100 and 110 each transmit a one. The satellite, on the basis of received signal strength, selects one or zero as the bit it "heard." In this example the satellite chooses binary one

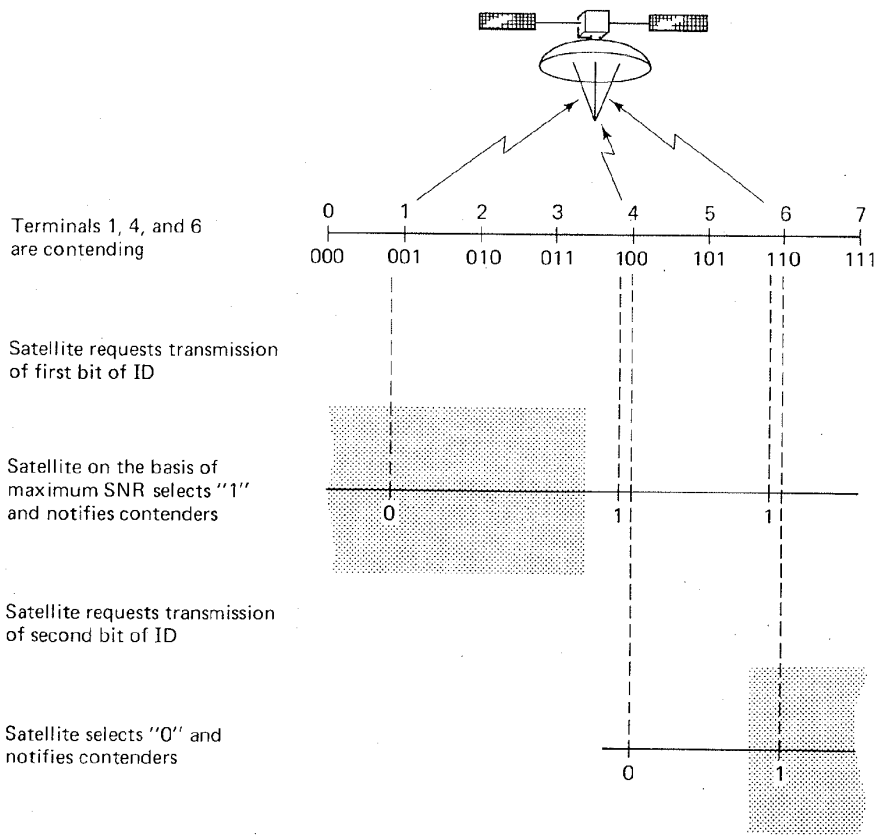


Figure 9.25 Tree search to resolve contention (eight terminal example).

and informs the users accordingly. Half the user population now knows that it has not been selected. The terminals in the "losing" half "bow out" of contention during this pass through the tree. In this example terminal 001 bows out.

3. The satellite requests the transmission of the second identifying bit from the remaining contending terminals.
4. Terminal 100 transmits a zero, and terminal 110 transmits a one.
5. Assume that the satellite selects the zero and notifies the contenders accordingly. Terminal 110 bows out. The process continues until it is clear that terminal 100 is free to access the satellite.
6. When the channel becomes available, steps 1–5 are repeated.

Example 9.3 Comparison between Binary Tree Search and Straight Polling

(a) A binary tree search requires $n = \log_2 Q$ decisions for each pass through a population of Q terminals. A savings in time is possible with a tree search if the population is large and the average demand for service is small. Calculate the

time needed for the straight polling of a population of 4096 terminals, to provide channel availability to 100 terminals requesting service. Compare the result with the time needed to perform a binary tree search 100 times, over the same population. Assume that the time required to poll one terminal and the time required for one decision of a binary tree search are each equal to 1 s.

(b) Develop an expression for Q', the largest number of terminals that results in the same (or less) time expended for binary tree searching as compared to straight polling.

(c) Compute Q' for part (a).

Solution

(a) Straight polling of 4096 terminals:

$$T = 4096 \times 1 \text{ s} = 4096 \text{ s}$$

Binary tree search for 100 terminals requires 100 passes through the binary tree:

$$T' = (100 \times \log_2 4096) \times 1 \text{ s} = 1200 \text{ s}$$

(b) Q' is the maximum number of terminals that will result in $T' \leq T$ in part (a). This will occur when

$$Q'' \log_2 Q \times 1 \text{ s/decision} = Q \times 1 \text{ s/poll}$$

$$Q' = \lfloor Q'' \rfloor = \left\lfloor \frac{Q}{\log_2 Q} \right\rfloor \tag{9.30}$$

where $\lfloor x \rfloor$ is the largest integer no greater than x.

(c) Q' for part (a)

$$Q' = \left\lfloor \frac{4096}{\log_2 4096} \right\rfloor = 341 \text{ terminals}$$

A binary tree search for 341 terminals entails a search time of 4092 s.

9.4 MULTIPLE ACCESS TECHNIQUES EMPLOYED WITH INTELSAT

The first commercial, geostationary communication satellite (INTELSAT I, or Early Bird) launched in 1965, represented the start of a new telecommunications era. Its 240 voice circuits provided more capacity than the undersea cables laid between the United States and Europe during the previous 10 years [15].

Early Bird featured a hard-limiting nonlinear transponder using FDMA. When several signals having different carrier frequencies simultaneously occupy a nonlinear device, the result is the production of intermodulation products which are signals at all combinations of sum and difference frequencies [16–18]. The energy apportioned to these intermodulation or IM products represents a *loss* in the useful signal energy. In addition, if these IM products appear within the bandwidth of other signals, the effect is that of added *noise* for the other signals.

The nonlinear transponder in Early Bird allowed for only two earth stations (one in the United States and one in Europe) to simultaneously access the satellite.

Figure 9.26 illustrates this satellite's operation between the United States and Europe. Three European earth stations were interconnected via a terrestrial network. Each month a different European station accessed the satellite and distributed the traffic to the other two stations.

9.4.1 Preassigned FDM/FM/FDMA or MCPC Operation

INTELSAT II and III improved multiple access capability by operating their travelling-wave tube amplifiers (TWTA) in the linear region instead of the hard-limiting region. This kept the IM products at an acceptable level, allowing more than two simultaneous accesses. (The price paid was a reduction in power amplifier efficiency.) Thus many FM carriers from various earth stations could simultaneously access these satellites. The operation is designated preassigned multidestination FDM/FM/FDMA or simply FDM/FM, or multichannel per carrier (MCPC), and is illustrated in Figure 9.27. Long-distance calls originating in country A enter the telephone exchange and are multiplexed into a supergroup (five groups of 12 voice circuits each). Country A transmits the supergroup on a single FM carrier at frequency f_A. Each group within the supergroup has been preassigned to an earth station in country A for telephone traffic destined to countries

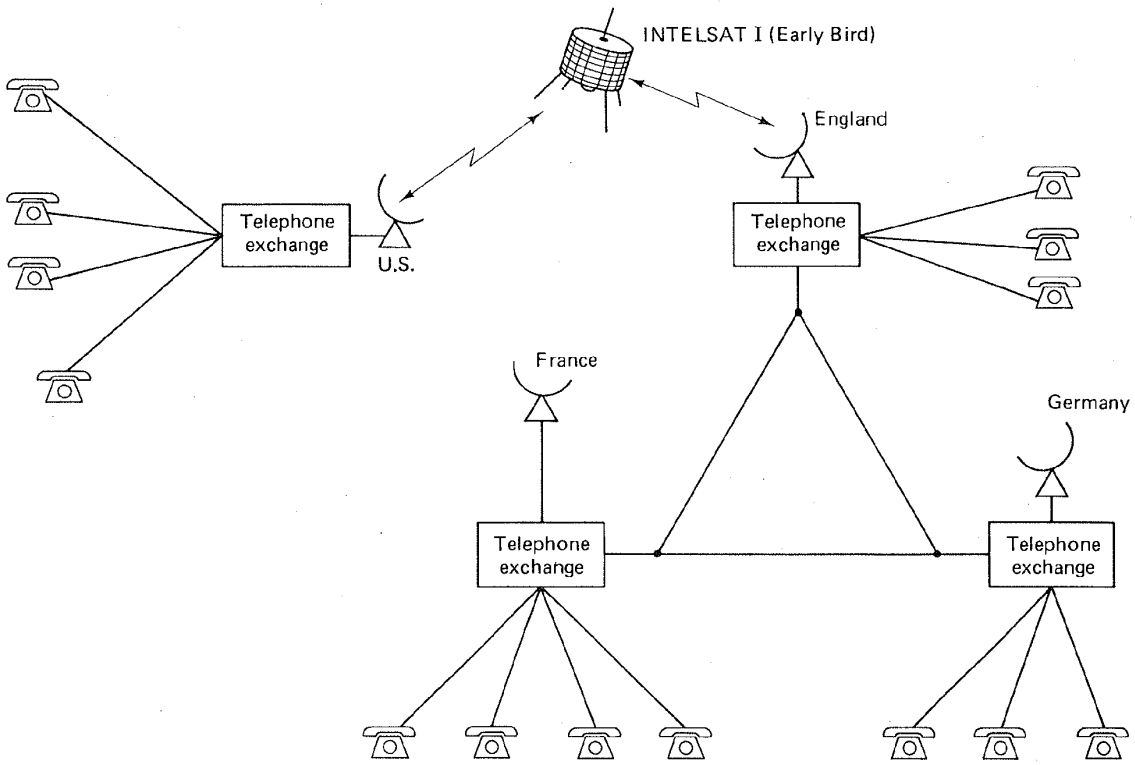


Figure 9.26 Early satellite operation.

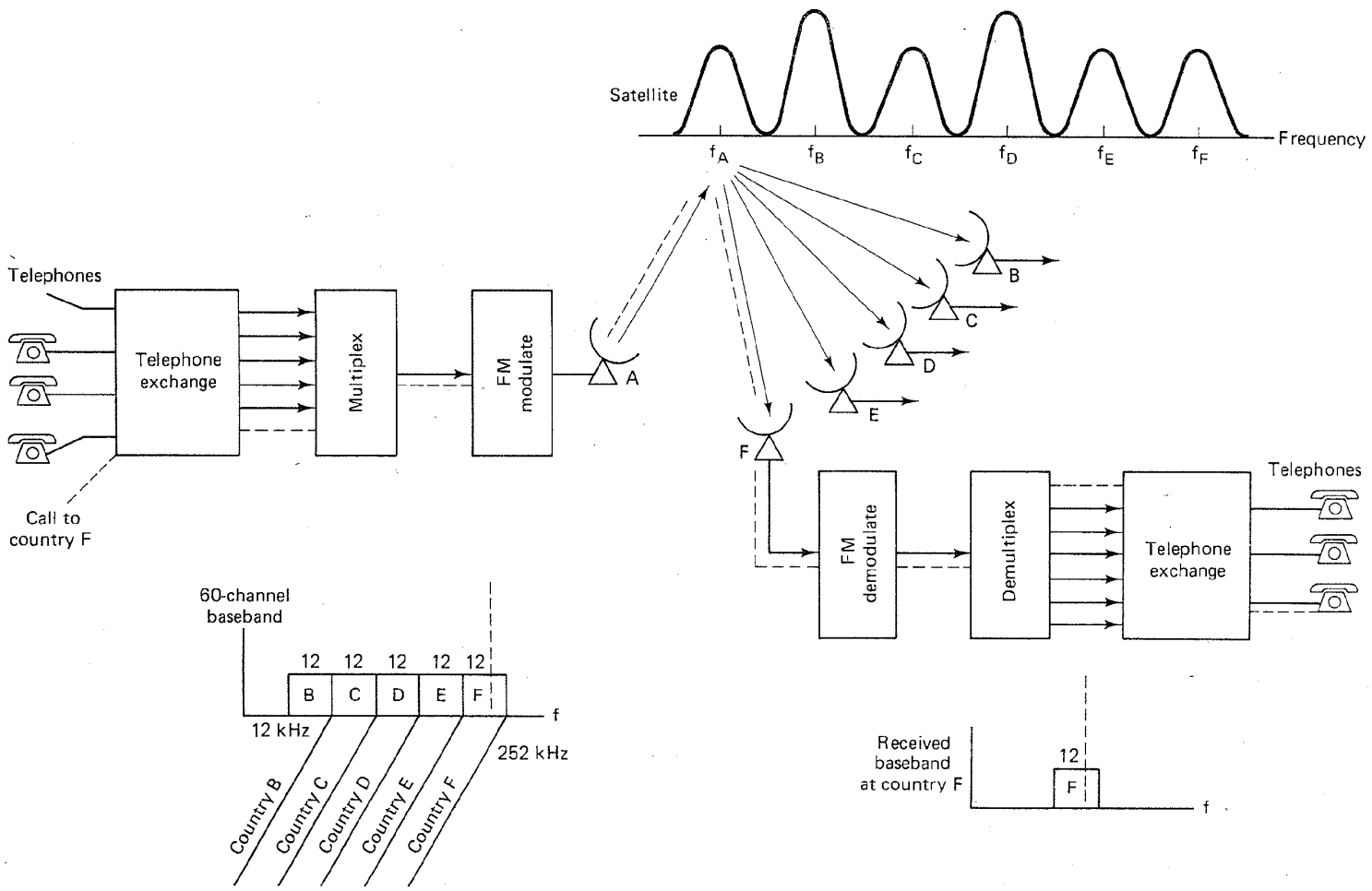


Figure 9.27 Preassigned multidestination FDM/FM carriers. (Reprinted with permission from J. G. Puente and A. M. Werth, "Demand-Assigned Service for the INTELSAT Global Network," *IEEE Spectrum*, Jan. 1971. © 1971 IEEE.)

B through F. These countries each receive the signal on frequency f_A. The received signal is demodulated and demultiplexed at the destination country, selecting only those 12 channels preassigned to it.

9.4.2 MCPC Modes of Accessing an INTELSAT Satellite

INTELSAT has standardized the ways in which each 36-MHz transponder may be shared by specifying the occupied RF bandwidth and the number of 4-kHz channels per user. Some of these standard channels are shown in Table 9.1. Notice that the capacity of the transponder (last column in Table 9.1) drops as the number of carriers increases. The reasons are as follows:

1. Guard bands are needed between carrier bands; the more carriers there are, the more guard bands are needed. Hence capacity is reduced.
2. Multiple carriers in the nonlinear TWTA cause intermodulation (IM) products. If the TWTA is backed off into the linear region to reduce interference, the TWTA can provide less overall power. The channel becomes power limited and can service fewer carriers.

TABLE 9.1 Standard INTELSAT MCPC Accessing Modes

Number of carriers per transponder	Carrier bandwidth	Number of 4-kHz channels per carrier	Number of 4-kHz channels per transponder
1	36 MHz	900	900
4	3 at 10 MHz	132	456
	1 at 5 MHz	60	
7	5 MHz	60	420
14	2.5 MHz	24	336

Table 9.1 indicates that a single carrier provides the most efficient use of the transponder. Why doesn't INTELSAT always operate its transponders in this mode? The answer is that not all earth stations have enough traffic to justify the assignment of an entire 36-MHz transponder. The other modes are needed so that various combinations of stations having less traffic will be able to share a transponder.

9.4.2.1 Bandwidth-Limited versus Power-Limited Conditions

In the preceding section it was stated that the backed-off transponder cannot support as many channels as the fully saturated transponder. It is useful to examine the two extreme transponder conditions, bandwidth limited and power limited, in the context of a satellite transponder. In Figure 9.28 we assume a 36-MHz transponder with a maximum power output of 20 W. Figure 9.28a illustrates an MCPC mode of operation whereby four carriers share the 36-MHz bandwidth. Assume that each carrier requires 4 W. The total output power is 16 W (less than the maximum capability of the amplifier); therefore, there is still power to spare.

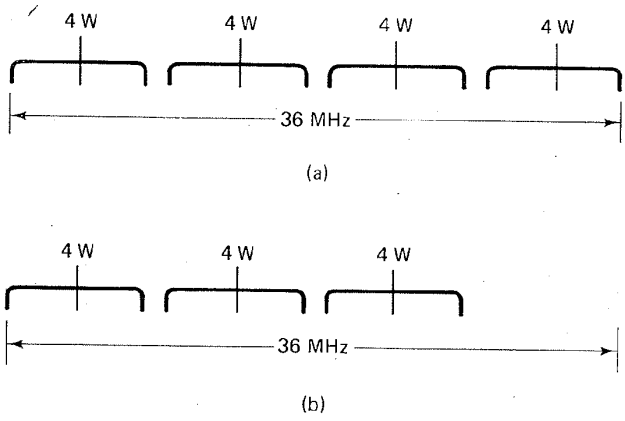


Figure 9.28 Bandwidth-limited versus power-limited configurations. (a) Bandwidth-limited example. (b) Power-limited example.

However, should another user want to access the transponder, the total 36-MHz bandwidth has already been allocated to the existing four carriers; there is no additional bandwidth to spare. Figure 9.28a illustrates this bandwidth-limited case.

Suppose that the previous example results in the production of serious IM products at the transponder. Assume that it is necessary to linearize the transponder by operating it at a reduced maximum power output of 12 W. With only a 12-W capability, the transponder can no longer support four users with 4 W each. One of the users must be "thrown off," as illustrated in Figure 9.28b. Therefore, we have bandwidth to support another user, but not sufficient power. Figure 9.28b illustrates this power-limited case.

9.4.3 SPADE Operation

The preassigned MCPC multiple access scheme is very efficient when the traffic is heavy enough so that the channels are most always filled. However, if out of a 12-channel group, only one channel is active, the other 11 cannot be turned off. The FDM/FM transmission is made with or without actual telephone traffic on the channels. Therefore, the long-term preassignment of carriers to stations having light traffic is wasteful. Since there are many light traffic links, a flexible method to service them was needed. Also, an efficient way to handle overflow traffic from medium-capacity preassigned links was needed. Such was the motivation for a novel DAMA scheme known as SPADE, first used with INTELSAT IV. The acronym SPADE stands for "single-channel-per-carrier PCM multiple access demand assignment equipment." The principal features characterizing SPADE operation [15] are:

1. A single voice-grade channel is analog-to-digital (A/D) converted at a bit rate of 64 kbits/s.
2. This baseband digital signal modulates a carrier using quadrature phase shift keying (QPSK). Unlike the MCPC case, there is *only one* voice channel per carrier.

3. The channel spacing is 45 kHz. Within a transponder, there is bandwidth available for 800 channel carriers. Six carrier positions are vacant by design; thus there are 794 usable carriers.
4. The carrier is dynamically assigned, *upon demand.*
5. The dynamic assignment is accomplished over a 160-kHz common signaling channel (CSC) used as an "order-wire" or control circuit. The bit rate on the CSC is 128 kbits/s, and the modulation is binary phase shift keying (BPSK).

Figure 9.29 illustrates the frequency allocations for the CSC and the 800 carriers in the SPADE system. The SPADE operation can best be understood with the aid of Figure 9.30. The CSC operates in a fixed-assignment TDMA broadcast mode; that is, all earth stations monitor the CSC and are aware of the current state of channel assignments. Each earth station has a 1-ms time slot on the CSC (once every 50 ms) for requesting or releasing a channel. When an earth station needs a channel, it "seizes" a free one by requesting a frequency pair at random and transmitting its selection on the CSC. Random selection makes it unlikely that two stations will simultaneously request the same channel unless there are very few remaining. As soon as the channel is allocated, each of the other earth station processors deletes it from its list of available channels. The list is continually kept updated via the CSC. Thus control of the SPADE access scheme is *distributed* among all the participating earth stations.

When the station finishes with the channel, the station indicates the channel's release by transmitting a signal in its time slot on the CSC. Each station receives this signal and designates the released channel as available. If two stations simultaneously seize the same channel, they each get a "busy" indication. They try again, selecting at random from the pool of available channels.

9.4.3.1 Transponder Capacity Utilization with SPADE

Table 9.2 is a continuation of Table 9.1. We see that the transponder bandwidth utilization with SPADE results in a total capacity of 800 voice channels per transponder. Compare Table 9.2 with Table 9.1. In Table 9.1, as the number of carriers increases from 1 to 14, the total number of channels decreases from 900 to 336. Why doesn't the SPADE system in Table 9.2 exhibit less capacity than the 336 channels associated with 14 carriers? The improved utilization comes about as follows. When there is only one voice channel per carrier, the carrier can be switched off when no speech is detected. Even with all channels operating, they can be switched off approximately 60% of the time. The transponders are power limited; power savings means that more channels can be transmitted. Also, SPADE uses digital voice transmission (QPSK); the bandwidth efficiency of the system is commensurate with the single-carrier FDM/FM case.

9.4.3.2 SPADE Efficiency

With MCPC, capacity is preassigned; a station's unused channels cannot be reallocated to other stations. SPADE is a DAMA system where all channels are shared. The channels are allocated to users as needed. An important telephone

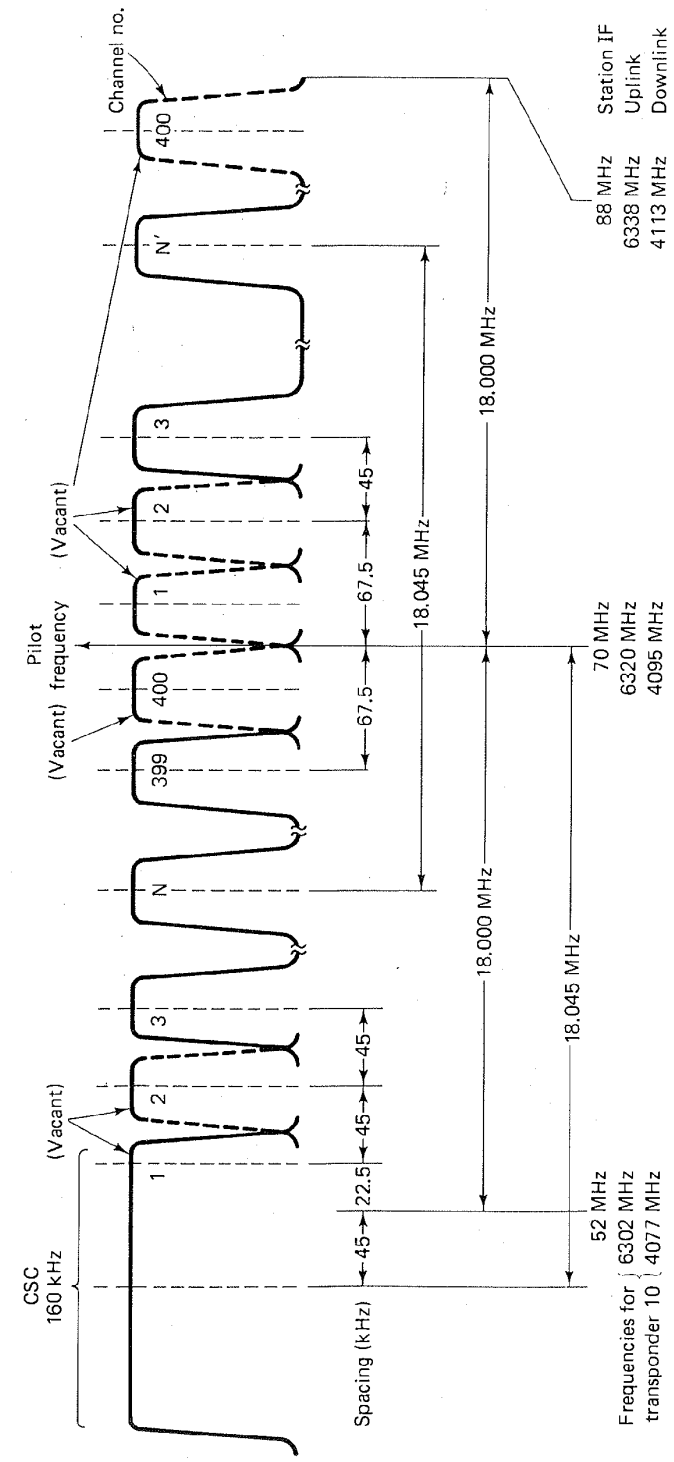


Figure 9.29 SPADE frequency allocations. (Reprinted with permission from J. G. Puente and A. M. Werth, "Demand-Assigned Service for the INTELSAT Global Network," *IEEE Spectrum*, Jan. 1971. © 1971 IEEE.)

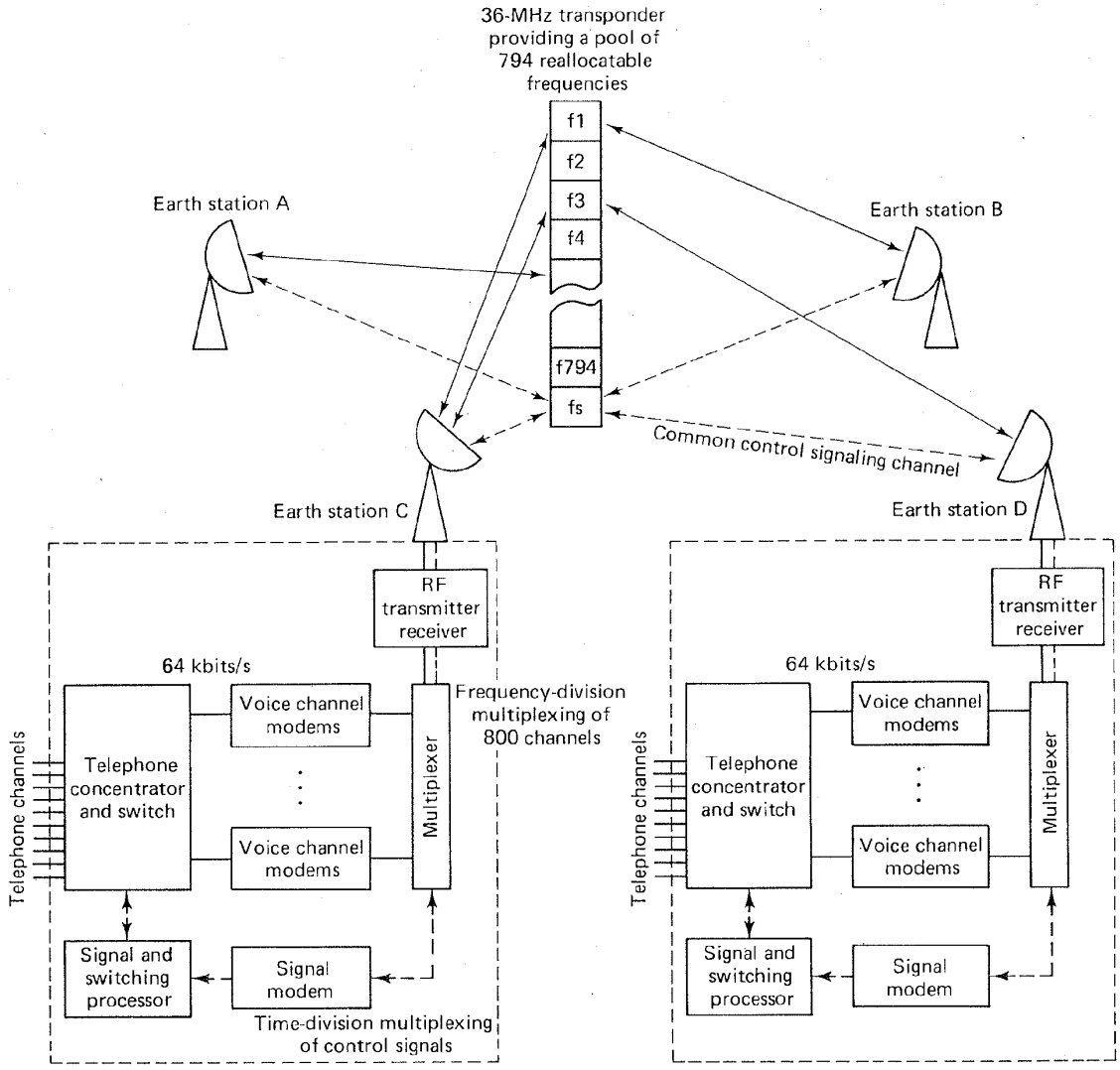


Figure 9.30 SPADE operation. (From James Martin, *Communications Satellite Systems*, © 1978, Fig. 15.2, p. 236. Reprinted by permission of Prentice-Hall, Englewood Cliffs, N.J.)

TABLE 9.2 SPADE Accessing

Number of carriers per transponder	Carrier bandwidth	Number of 4-kHz channels per carrier	Number of 4-kHz channels per transponder
800	45 kHz	1	800

system quality measure, called the probability of blocking, is the probability that a requested circuit is not available. To achieve 1% probability of blocking requires four times as many MCPC channels as SPADE channels. A SPADE transponder with 800 channels is equivalent to 3200 MCPC channels [15].

9.4.3.3 Mixed-Size Earth Station Network Using SPADE

A standard-size INTELSAT earth station has a receiver sensitivity G/T° = 40.7 dB/K, whereas the smaller size stations have a G/T° = 35 dB/K. If 125 SPADE channels are destined for small stations, the total transponder capacity of 800 standard channels is reduced to 525 channels. This is the point at which half the available power is used to service the standard stations. The relationship between transponder capacity and channels allocated to small stations is shown in Figure 9.31. An explanation of this relationship can best be seen in Figure 9.32. When the total TWTA power provides service to large stations, Figure 9.32a illustrates that the 36-MHz bandwidth transponder is occupied by approximately 800 carriers each at a power level of x dBW (the bandwidth-limited case). When half the power is required to service small stations, Figure 9.32b illustrates that 400 carriers (half of the original 800) each at a power level of x dBW are reserved for the standard stations. Consider what happens to the remaining 400 carrier positions. From Chapter 4 we know that the error performance of a link is directly related to the product of EIRP and G/T°. For any link, one can trade off these two parameters, thereby maintaining a fixed level of performance. Since the small station has a G/T° of 5.7 dB less than that of the standard station, it is necessary to supply the small station with 5.7 dB more EIRP for equivalent performance. The carrier power is *increased* by approximately 5.7 dB for each small station, thus the quan-

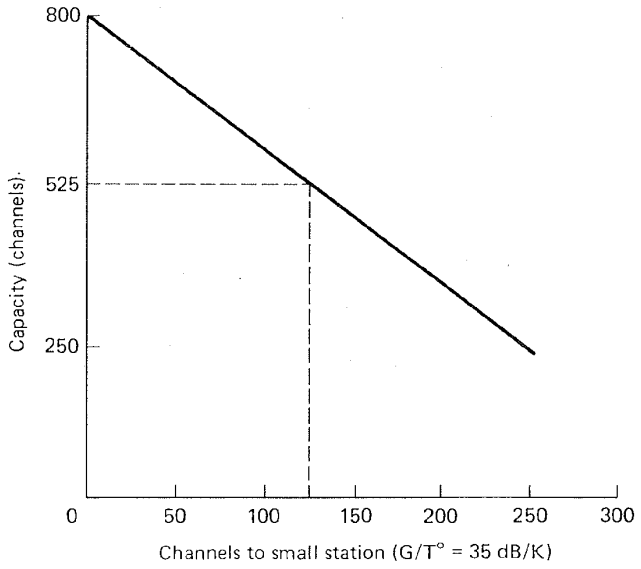


Figure 9.31 SPADE transponder capacity in a mixed-size earth station network.

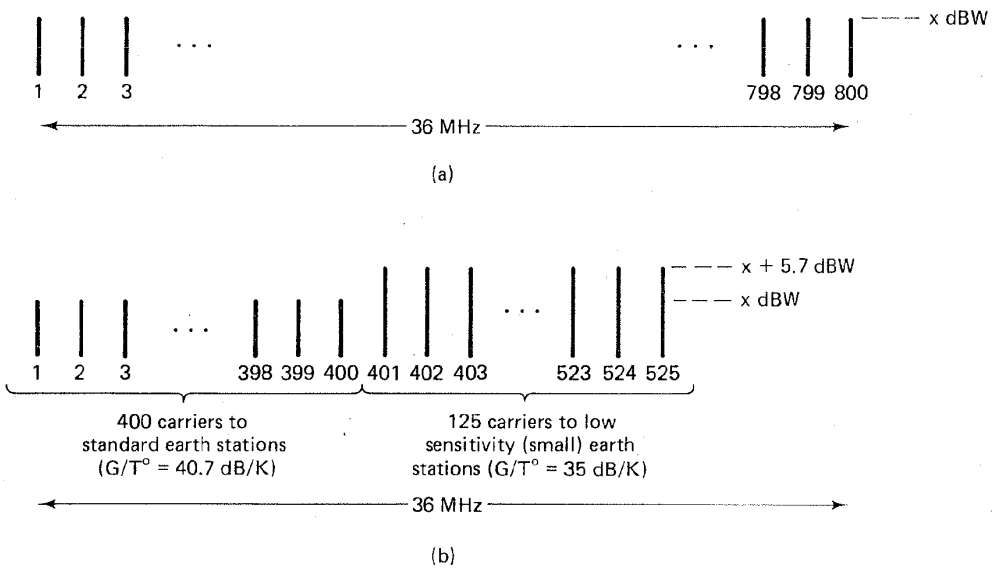


Figure 9.32 Mixed-size earth station network. (a) When the total TWTA power services large stations: bandwidth limited (800 carriers). (b) When half the TWTA power services small stations: power limited (525 carriers).

tity of the remaining carriers serving these small stations is *decreased* by a similar amount. Therefore instead of 400 carriers, 125 (a reduction of 5.1 dB) are used to serve the small stations; the transponder is now power limited.

At the time a channel is assigned to a call, the transmitting station is apprised of the size of the destination station. Recall that these satellites are nonregenerative so that the apportionment of downlink EIRP is established by the transmitting station (see Section 4.7.1). The transmitting station sets its power level according to the needs of the receiving station.

9.4.4 TDMA in INTELSAT

The first generation of multiple access communication systems has been dominated by FDMA systems. The trend, however, is now in favor of TDMA systems, made possible by the availability of precise clocks and high-speed switching elements [19–24]. INTELSAT IV used a 128-kbits/s TDMA scheme for the common signaling channel that controls the SPADE network. Intelsat V introduced a 120-Mbits/s TDMA scheme for multiple-beam international digital service. One disadvantage or cost in implementing a TDMA scheme is the need for providing precise *synchronization* among the participating earth stations and the satellite. FDMA systems, not having such requirements, are less complex from a networking point of view. Comparisons of TDMA versus FDMA operation are summarized as follows:

1. FDMA can cause IM products. This can be avoided by operating the TWTA in its linear region, thereby reducing the available power output.
2. With TDMA, there is only one carrier present at a time in the TWTA. Thus IM distortion cannot occur.
3. TDMA earth station equipment is more sophisticated and hence more costly than FDMA equipment. However, for earth stations providing multiple point-to-point channels, FDMA stations require separate radio-frequency (RF) up-conversion and down-conversion signal processing stages. Thus with FDMA, the amount of equipment grows with the amount of simultaneous connectivity. With TDMA, such growth does not take place since channel selectivity is accomplished in time rather than frequency. Therefore, for a large multiply connected earth station, TDMA can be more cost-effective than FDMA.
4. In multiple-beam systems, each beam may need to communicate with every other beam. TDMA lends itself to conveniently forming connections sequentially as in satellite-switched TDMA (SS/TDMA). INTELSAT VI uses such satellite-switched TDMA (SS/TDMA), described in Section 9.4.5.

An example of the comparative performance of TDMA, FDM/FM, and SPADE is shown for an INTELSAT IV transponder as a plot of channel capacity versus earth station G/T° in Figure 9.33. Figure 9.33a is for an earth coverage antenna, and Figure 9.33b is for a spot-beam antenna. From synchronous altitudes these antennas have half-power beamwidths of 17° and 4.5°, respectively. From these plots it is seen that single-carrier FDM/FM is as efficient as TDMA when the system is operated with standard earth stations (G/T° = 40.7 dB/K). For smaller earth stations ($G/T^\circ \leq$ 31 dB/K) working through earth-coverage transponders, SPADE is more efficient than TDMA and multicarrier FDM/FM (MCPC); only the four-carrier case is plotted. For earth stations having G/T° in the range 19 to 40.7 dB/K working through a spot-beam transponder, TDMA is superior to SPADE and MCPC. For smaller earth stations having G/T° in the range 6 to 19 dB/K working through a spot-beam transponder, SPADE is superior to TDMA and MCPC. In general, when working through *standard* earth stations it is seen [19] that TDMA is the most efficient multiple access scheme for INTELSAT IV.

9.4.4.1 PCM Multiplex Frame Structures

There are two digital telephony standards for PCM frame structures in operation. The North American standard is called *T-Carrier*; it is built around the 193-bit frame shown in Figure 9.34a. There are 24 channels; each channel contains an 8-bit voice sample. Also, there is one bit per frame with alternating value 1 0 1 0 . . . from frame to frame, used for frame alignment. Since a voice-grade telephone channel has a bandwidth of W = 4 kHz (including guard bands), the Nyquist sampling rate for recovering the analog information within 4 kHz is $f_s = 2W$ = 8000 samples/s. Therefore, the basic PCM frame, called the *Nyquist*

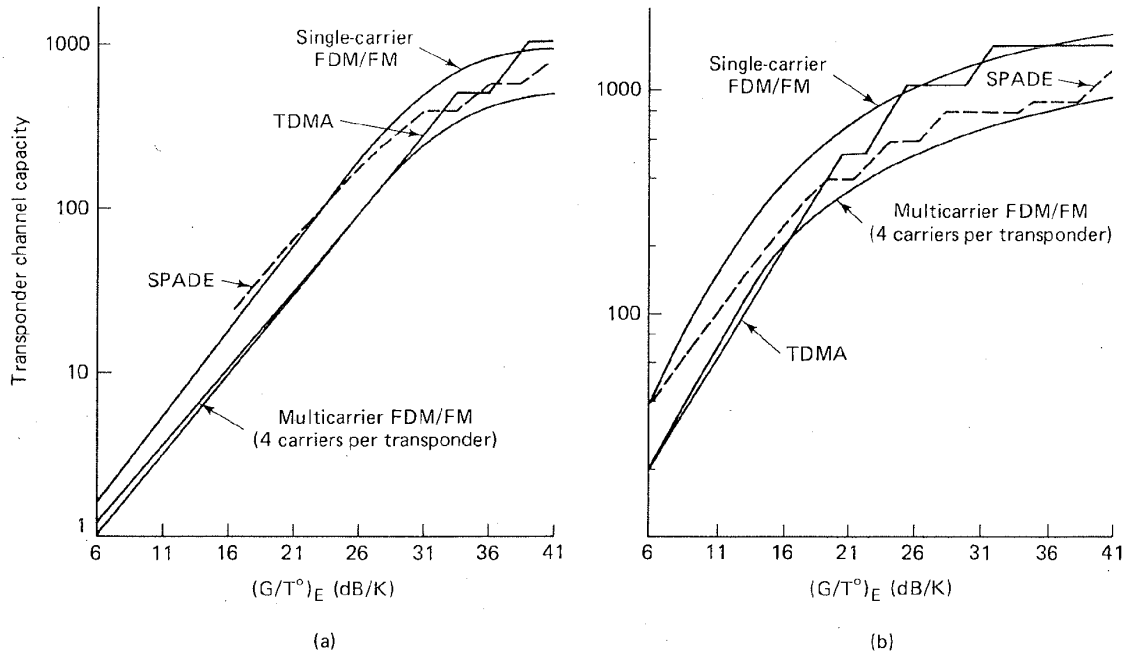


Figure 9.33 Channel capacity versus earth station G/T° for FDMA, TDMA, and SPADE. (a) Global-beam transponder channel capacity as function of $(G/T^\circ)_E$, where $(G/T^\circ)_E$ means earth station G/T°. (b) Spot-beam transponder channel capacity as function of $(G/T^\circ)_E$. [From D. Chakraborty, "INTELSAT IV Satellite System (Voice) Channel Capacity versus Earth Station Performance," *IEEE Trans. Commun. Tech.*, vol. COM19, no. 3, June 1971, pp. 355–362. © 1971 IEEE.]

frame, which contains 24 voice samples from 24 different message sources, has a frame rate of 8000 frames/s (duration of 125 μs). Thus the basic T-Carrier bit rate is 193 bits/frame × 8000 frames/s = 1.544 Mbits/s.

The European standard is built around a 256-bit frame shown in Figure 9.34b. There are 30 message channels, each containing an 8-bit voice sample. Also, one 8-bit time slot is used for frame alignment and another 8-bit time slot is used for signaling (addressing) information. The European frame rate is the same as that of the T-Carrier. Therefore, the basic European bit rate is 256 bits/frame × 8000 frames/s = 2.048 Mbits/s.

9.4.4.2 The High-Rate TDMA Frame for Europe

Sixteen Nyquist frames of the European PCM Multiplex format are shown in Figure 9.35a. Each frame contains an 8-bit sample from each of 30 terrestrial channels, plus 8 bits of framing and 8 bits of signaling information. The TDMA frame duration is

$$16 \text{ Nyquist frames} \times 125 \text{ μs/Nyquist frame} = 2 \text{ ms}$$

Within this 2-ms frame are contained

$$16 \text{ Nyquist frames} \times 256 \text{ bits/Nyquist frame} = 4096 \text{ bits}$$

The basic idea behind TDMA is that a user's low-rate data stream can share the CR with similar streams from other users by *bursting* the transmission at a much faster rate than the rate at which it is generated. Figure 9.35b illustrates a 2-ms high-rate TDMA frame. The frame begins with a reference burst, RB1, emitted by a reference station. The burst contains information necessary to enable other stations to precisely position their message traffic bursts in the frame. There may be a second burst, RB2, for reliability, followed by a sequence of traffic slots. The traffic slots may be preassigned, or they may be assigned according to a DAMA protocol [20].

The PCM multiplex signal with a bit rate of $R_0 = 2.048$ Mbits/s and a frame duration of $T = 2$ ms is compressed (by a factor of 59) and transmitted using QPSK modulation at a burst rate of $R_T = 120.832$ Mbits/s (symbol rate of 60.416 megasymbols/s). The duration of the traffic data field T_{tr} in the high rate TDMA

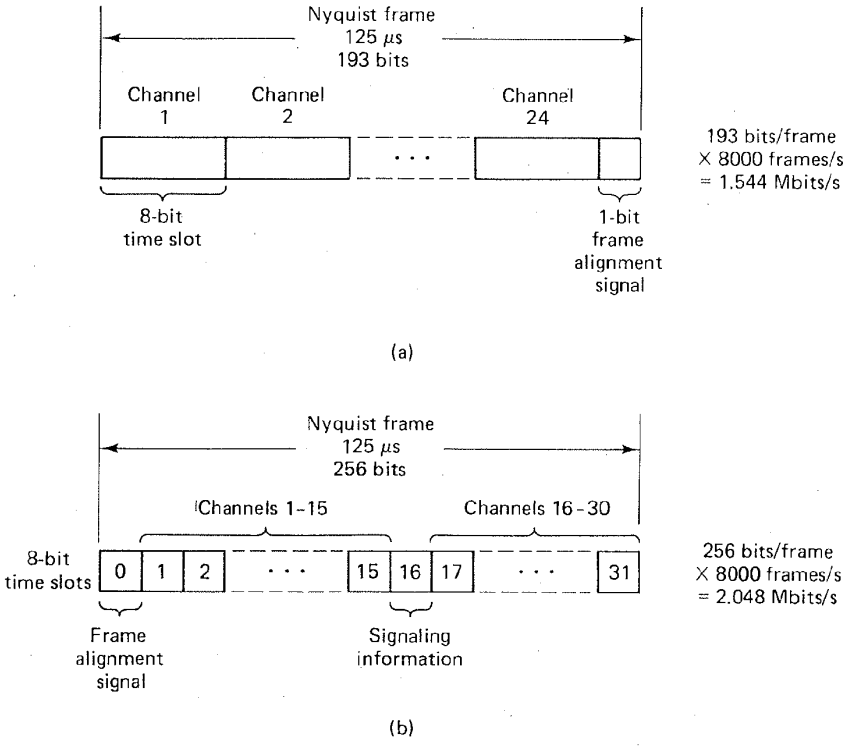


Figure 9.34 PCM multiplex frame structure. (a) Frame structure for T-Carrier (North American) PCM multiplex. (b) Frame structure for the European PCM multiplex.

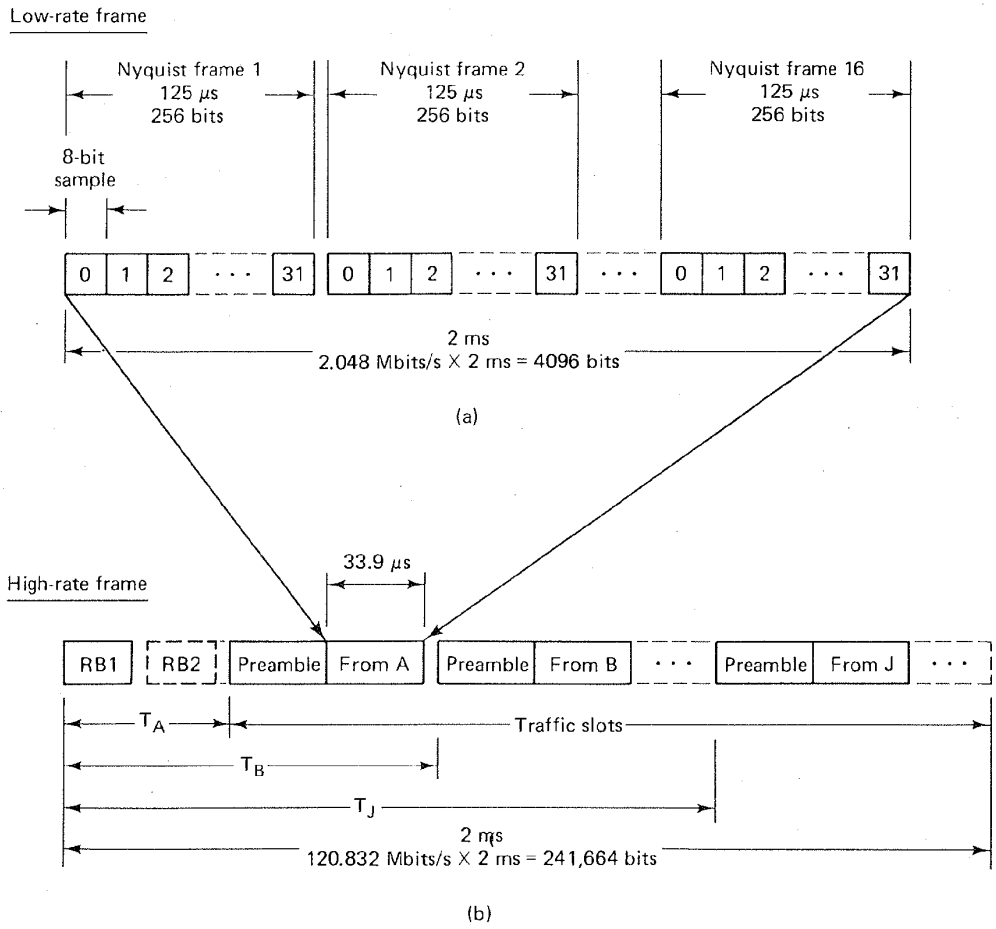


Figure 9.35 INTELSAT digital transmission standards for Europe. (a) Terrestrial PCM multiplex. (b) High-rate frame.

frame is calculated as follows:

$$T_{\text{tr}} = \frac{R_0 T}{R_T} \tag{9.31}$$

$$= \frac{2.048 \times 10^6 \times 2 \times 10^{-3}}{120.832 \times 10^6}$$

$$= 33.9\ \mu\text{s}$$

To obtain the total duration of a traffic burst, the time used for the preamble must be added. If the preamble contains S_P symbols, then assuming QPSK modulation, the total length of the traffic burst measured in number of symbols, S_T, is

$$S_T = \frac{R_0 T}{2} + S_P \tag{9.32}$$

and the burst-time duration is

$$T_T = \frac{2S_T}{R_T} \tag{9.33}$$

If the preamble contains 300 symbols, then

$$S_T = \frac{2.048 \times 10^6 \times 2 \times 10^{-3}}{2} + 300$$

$$= 2348 \text{ symbols}$$

Using this in Equation (9.33), we obtain

$$T_T = \frac{2 \times 2348}{120.832 \times 10^6} = 38.9 \ \mu\text{s}$$

9.4.4.3 The High-Rate TDMA Frame for North America

The INTELSAT TDMA burst (bit) rate of $R_T = 120.832$ Mbits/s was chosen to be compatible with both the European and North American standards. Figure 9.36 is similar to Figure 9.35 except that the PCM multiplex signal is the 24-channel T-Carrier instead of the 30-channel European standard. The essential T-Carrier features that are different from the European standard are listed below and are shown on the figure.

1. Each Nyquist frame is comprised of 24 channels or samples $\times$ 8 bits + 1 frame alignment bit = 193 bits.
2. The 16 Nyquist frames contain $16 \times 193 = 3088$ bits.
3. The T-Carrier data rate is 1.544 Mbits/s.
4. The duration of the traffic data field in the high-rate TDMA frame is calculated from Equation (9.31).

$$T_{\text{tr}} = \frac{1.544 \times 10^6 \times 2 \times 10^{-3}}{120.832 \times 10^6}$$

$$= 25.6 \ \mu\text{s}$$

9.4.4.4 INTELSAT TDMA Operation

At the transmitting earth station, the continuous low-rate data stream enters one of a pair of buffers illustrated in Figure 9.37a. When one buffer is filling at the low rate (1.544 Mbits/s or 2.048 Mbits/s), the other is emptying at the burst rate (120.832 Mbits/s). The buffers alternate functions at each TDMA frame. The time of application of the high-rate clock is controlled so that the traffic burst is transmitted in the proper interval to arrive at the satellite in its assigned position in the TDMA frame.

At the receiving station, the received traffic burst is routed to one of a pair of expansion buffers, shown in Figure 9.37b, that have the inverse function of

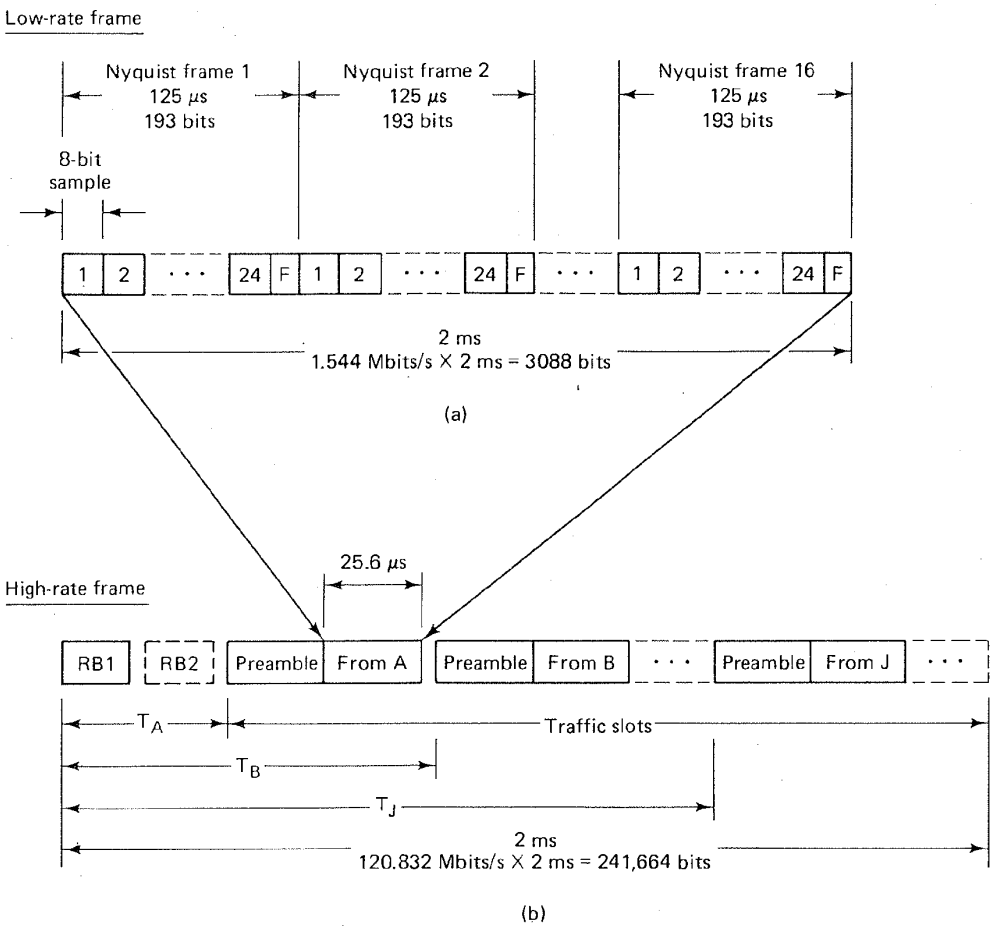


Figure 9.36 INTELSAT digital transmissions standards for T-Carrier. (a) Terrestrial PCM multiplex. (b) High-rate frame.

the compression buffers in Figure 9.37a. When one buffer is filling at the high rate, the other is emptying at the desired output rate.

The most critical aspect of TDMA operation is the precise synchronization needed to assure orthogonality of the time slots [20]. Figure 9.38 illustrates the general idea behind most commercial satellite synchronization schemes. One station is designated as the master or control station. This station transmits periodic bursts of reference timing pulses. User stations also transmit their timing pulses, designated as slave pulses in Figure 9.38. On the downlink, the using station receives the master or reference pulses in addition to its own slave pulses. The time difference between the master and slave pulses corresponds to the timing error. The station adjusts its clock so as to reduce this timing error.

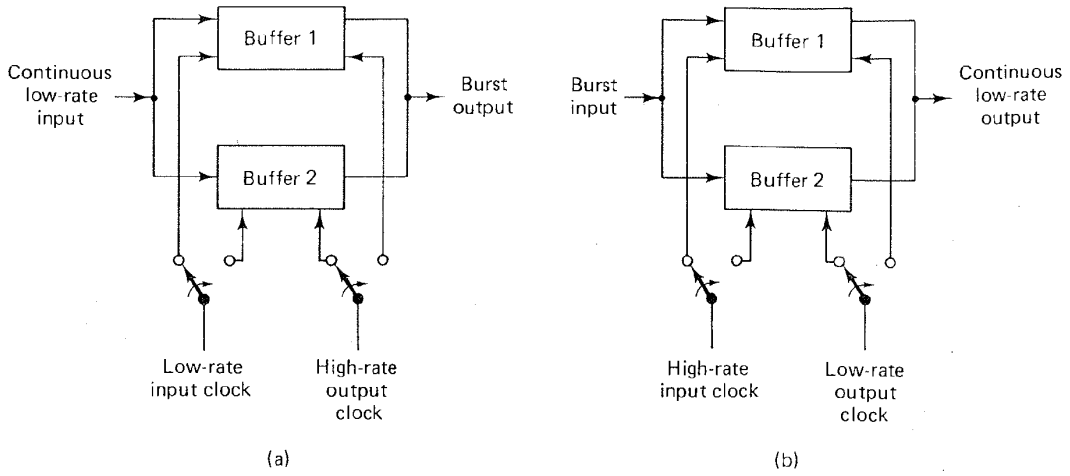


Figure 9.37 Burst compression and expansion buffers. (a) Compression buffers at transmitter. (b) Expansion buffers at receiver.

9.4.5 Satellite-Switched TDMA in INTELSAT

Modern communication satellites often employ several regional antenna beams. For a satellite based over the Atlantic Ocean, separate beams might be aimed at North America, Europe, South America, and Africa. Switches are used to allow the interconnection of stations in one region to communicate with stations in another region. The basic goal of a satellite-switched TDMA (SS/TDMA) scheme is to provide an efficient way of cyclically providing interconnection of TDMA data among various coverage regions.

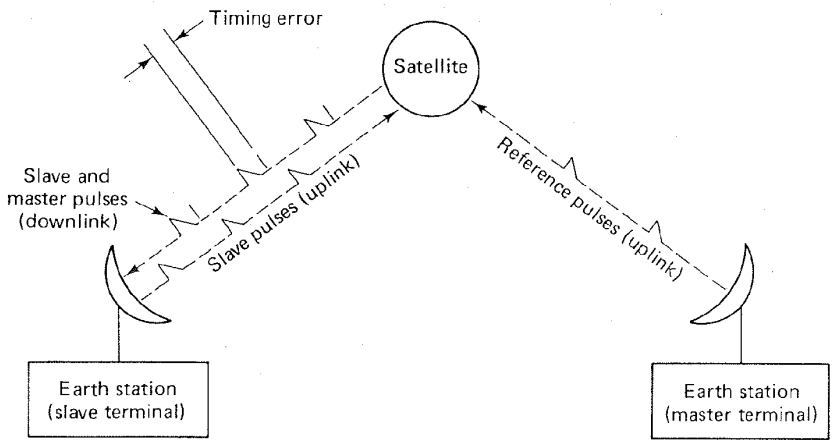


Figure 9.38 TDMA synchronization concept.

The heart of the system consists of a microwave switch matrix, located in the satellite, that is programmed via ground control to change states cyclically in rapid sequence, thus interconnecting distinct uplink beams to distinct downlink beams at each switching time. An earth station in the network communicates with those in other beams by transmitting TDMA bursts in the proper timing positions in the sequence. The pattern of switch states is selected so as to maximize the usable system capacity under the constraints of the traffic demands [21]. For complete interconnectivity between N beams, a total of $N!$ different satellite switch states or *modes* are required. Table 9.3 illustrates the six modes required for the full interconnectivity of a three-beam system.

In mode 1, the satellite receivers in beams A, B, and C are connected to the satellite transmitters for beams A, B, and C, respectively. An earth station in one of these beams can then communicate with other earth stations in the same beam. The beam is said to be *looped back* on itself.

Figure 9.39 illustrates a three-beam (beams A, B, and C) example of a SS/TDMA system. The satellite microwave switch matrix is configured in a *crossbar* design. This design can be thought of as being made up of row and column lines; when one row and one column are energized, contact is made at the intersection. A crossbar design only permits a single row to communicate with a single column at a time. If uplink A_U is connected to downlink B_D, *neither* A_U *nor* B_D can be simultaneously connected to any other beam.

In Figure 9.39, three different traffic patterns during time slot intervals T_1, T_2, and T_3, with three different switch states S_1, S_2, and S_3 are shown. During interval T_1, switch state S_1 interconnects the beams in a loop-back fashion which permits the uplink messages in slot T_1 to be delivered to their correct destinations. During time interval T_2, switch state S_2 interconnects uplink beam A_U to downlink beam B_D, uplink beam B_U to downlink beam C_D, and uplink beam C_U to downlink beam A_D. This connection pattern assures that the uplink messages in slot T_2 are delivered to their correct destinations. During time interval T_3, switch state S_3 similarly connects uplink transmissions to downlink beams to assure correct delivery of the data.

The traffic patterns and their durations are programmed to optimize the resource capacity and to serve the users as efficiently as possible. The cyclic pattern can be reprogrammed by ground command to meet changing traffic requirements.

9.4.5.1 Traffic Matrix

Figure 9.40 is a matrix describing the communication traffic among N spot-beam coverages. In this figure, t_{ij} is the traffic volume from the ith beam to the jth beam. The subtotal S_i is the total traffic originating from the ith uplink beam, expressed as

$$S_i = \sum_{j=1}^{N} t_{ij} \tag{9.34}$$

TABLE 9.3 Three-Beam Satellite Switch Modes

Input	Output					
	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
A	A	A	B	B	C	C
B	B	C	A	C	A	B
C	C	B	C	A	B	A

and R_j is the total traffic received in the jth downlink beam, expressed as

$$R_j = \sum_{i=1}^{N} t_{ij} \tag{9.35}$$

When the traffic in a SS/TDMA system is controlled by a nonblocking switch (one that allows for the transmission of *all* messages, without any "busy" signals)

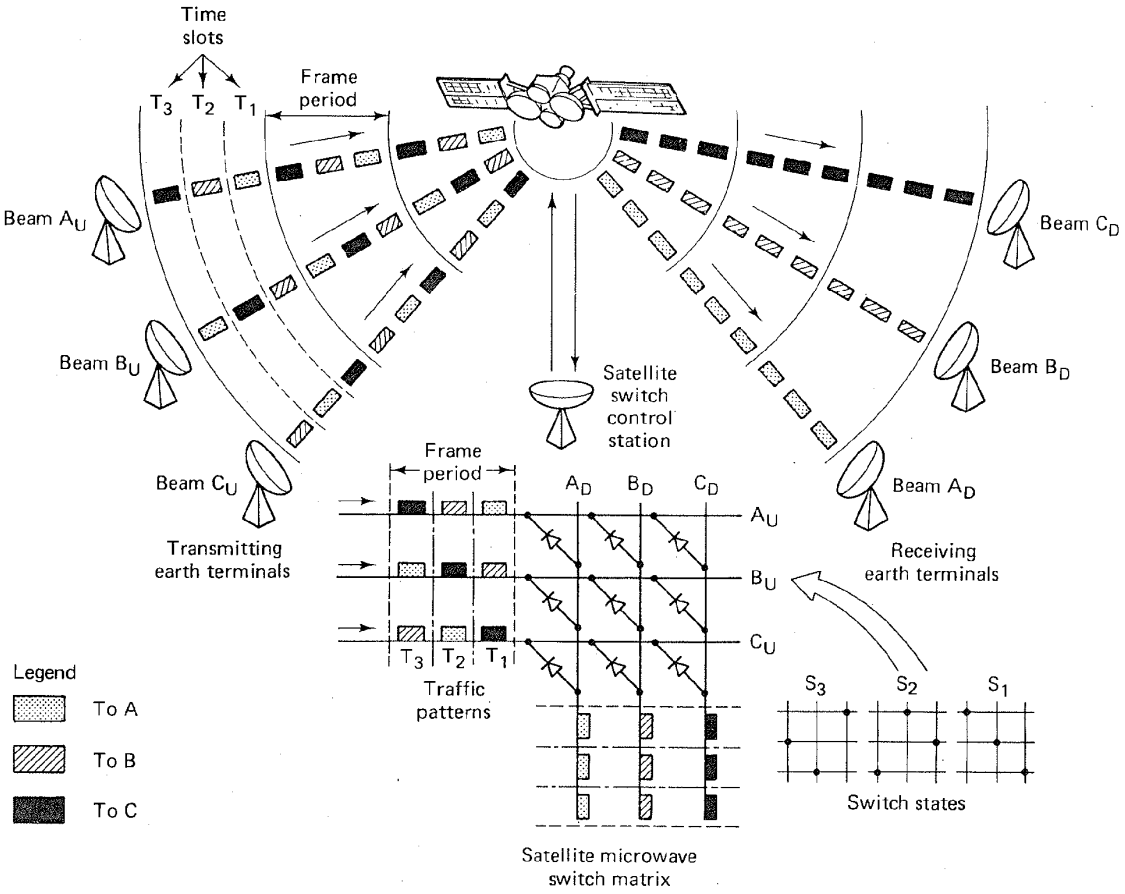


Figure 9.39 Satellite-switched TDMA (SS/TDMA).

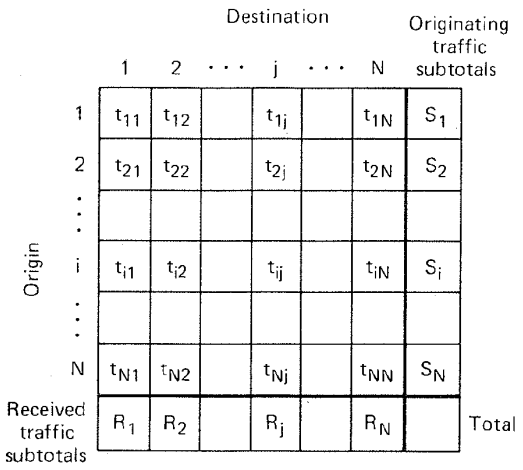


Figure 9.40 Traffic matrix.

a k-second time slot will be assigned to each channel in the TDMA frame. For efficient utilization of the CR, the total traffic in Figure 9.40 should be transmitted within a frame time T which should be made as short as possible. The minimum frame time, $T_{\min}$, for providing such nonblocking connectivity can be expressed [22] as follows:

$$T_{\min} = k \max (\{S_i\}, \{R_j\}) \tag{9.36}$$

where max $(\{S_i\}, \{R_j\})$ is the maximum value taken over the set of all $\{S_i\}$ and $\{R_j\}$. Equation (9.36) describes the minimum time to communicate *all* of the traffic in the traffic matrix, for equal bandwidth per channel.

9.5 MULTIPLE ACCESS TECHNIQUES FOR LOCAL AREA NETWORKS

A local area network (LAN) can be used to interconnect computers, terminals, printers, and so on, located within a building or a small set of buildings. While long-haul networks use the public telephone network for economic reasons, LAN designers usually lay their own high-bandwidth cables. Bandwidth is not as scarce as it is in the long-haul cases. Not being forced to optimize bandwidth, a LAN can use simple access algorithms [6, 25–27].

9.5.1 Carrier-Sense Multiple Access Networks

Ethernet is a LAN access scheme developed by the Xerox Corporation. The Ethernet scheme is based on the assumption that each local machine can sense the state of a common broadcast channel before attempting to use it. The technique is known as *carrier-sense multiple access with collision detection* (CSMA/CD). The word "carrier," here, means *any* electrical activity on the cable. Figure 9.41a

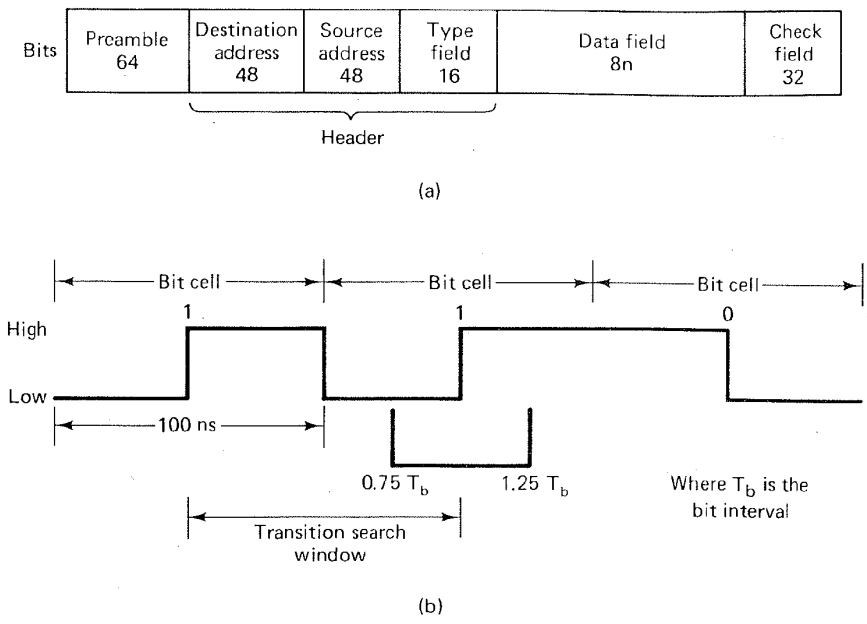


Figure 9.41 Ethernet bit field and PCM format. (a) Ethernet specification. (b) Manchester PCM format.

illustrates the bit field format for the Ethernet specification; the details are listed below.

1. The maximum packet size is 1526 bytes, where a byte is 8 bits. The packet breakdown is 8-byte preamble + 14-byte header + 1500-byte data + 4-byte parity.
2. The minimum packet size is 72 bytes, consisting of an 8-byte preamble + 14-byte header + 46-byte data + 4-byte parity.
3. The minimum spacing between packets is 9.6 μs.
4. The preamble contains a 64-bit synchronization pattern of alternating ones and zeros, ending with two consecutive ones: (1 0 1 0 1 0 . . . 1 0 1 0 1 1).
5. The receiving station examines a destination address field in the header to see if it should accept a particular packet. The first bit indicates the type of address (0 = single address, 1 = group address); an entire field of ones means an all-station broadcast.
6. The source address is the unique address of the transmitting machine.
7. The type field determines how the data field is to be interpreted. For example, bits in the type field can be used to describe such things as data encoding, encryption, message priority, and so on.
8. The data field is an integer number of bytes from a minimum of 46 to a maximum of 1500.

9. The parity check field houses the parity bits which are generated by the following generating polynomial (see Section 5.6):

$$X^{32} + X^{26} + X^{23} + X^{22} + X^{14} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$$

The Ethernet multiple access algorithm defines the following user action or response:

1. *Defer.* The user must not transmit when the carrier is present or within the minimum packet spacing time.
2. *Transmit.* The user may transmit if not deferring until the end of the packet or until a collision is detected.
3. *Abort.* If a collision is detected, the user must terminate packet transmission and transmit a short jamming signal to ensure that all collision participants are aware of the collision.
4. *Retransmit.* The user must wait a random delay time (similar to the ALOHA system) and then attempt retransmission.
5. *Backoff.* The delay before the nth attempt is a uniformly distributed random number from 0 to $2^n - 1$, for $(0 < n \leq 10)$. For $n > 10$, the interval remains 0 to 1023. The unit of time for the retransmission delay is 512 bits (51.2 μs).

Figure 9.41b illustrates a 10-Mbits/s data stream with Manchester PCM formatting from the Ethernet specification. Notice that with such formatting, each bit cell or bit position contains a transition. A binary one is characterized by transitioning from a low level to a high level, while a binary zero has the opposite transition. Therefore, the presence of data transitions denotes to all "listeners" that the carrier is present. If a transition is not seen between 0.75 and 1.25 bit times since the last transition, the carrier has been lost, indicating the end of a packet.

9.5.2 Token-Ring Networks

A carrier-sense network consists of a cable onto which all stations are passively connected. A *ring network,* by comparison, consists of a series of point-to-point cables between consecutive stations. The interfaces between the ring and the stations are active rather than passive. Figure 9.42a illustrates a typical unidirectional ring with interface connections to several stations. Figure 9.42b illustrates the state of the interface for the listen mode and the transmit mode. In the *listen mode* the input bits are copied to the output with a delay of one bit time. In the *transmit mode,* the connection is broken so that the station can enter its own data onto the ring. The token is defined as a special bit pattern (e.g., 1 1 1 1 1 1 1 1) which circulates on the ring whenever all stations are idle. How does the system ensure that message data do not contain a tokenlike sequence? *Bit stuffing* is used to prevent this pattern from occurring in the data. For the

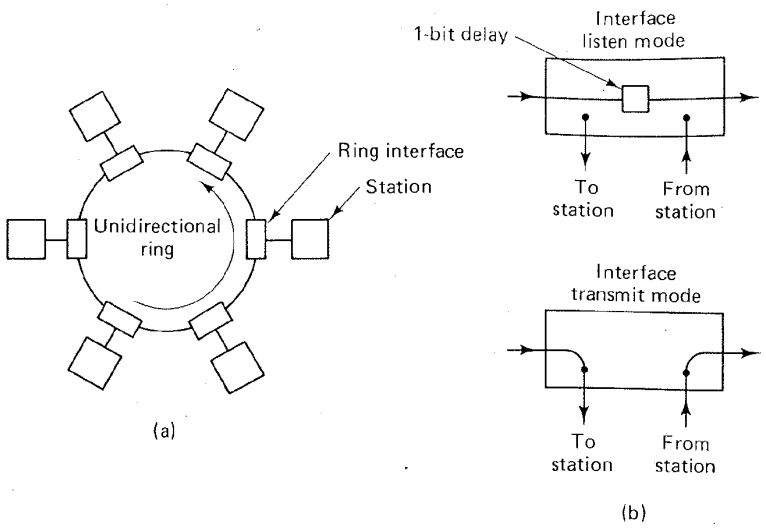


Figure 9.42 Token-ring network. (a) Network. (b) Listen and transmit modes.

8-bit token example shown, a bit-stuffing algorithm would insert a zero into the data stream after each sequence of seven consecutive ones. The data receiver would use a similar algorithm to dispose of the inserted bit following any sequence of seven consecutive ones. The token-ring access scheme works as follows:

1. A station wanting to send a message monitors the token appearing at the interface. When the last bit of the token appears, the station inverts it (e.g., 1 1 1 1 1 1 1 0). The station then breaks the interface connection and enters its own data onto the ring.
2. As bits come back around the ring, they are removed by the sender. There is no limit on the size of the packets, because the entire packet never appears on the ring at one instant.
3. After transmitting the last bit of its message, the station must regenerate the token. After the last data bit has circled the ring and has been removed, the interface is switched back to the listen mode.
4. Contention is not possible with a token-ring system. During heavy traffic, as soon as a token is regenerated, the next downstream station requiring service will see and remove the token. Thereby, permission to transmit rotates smoothly around the ring. Since there is only one token, there is no contention.

The ring itself must have sufficient delay to enable a complete token to circulate when all stations are idle. A major issue in ring network design is the propagation distance or "length" of a bit. If the data rate is R Mbits/s, a bit is emitted every $(1/R)$ microseconds. Since the propagation rate along a typical coaxial cable is 200 m/μs, each bit occupies $200/R$ meters on the ring.

Example 9.4 Minimum Ring Size

If an 8-bit token is to be used on a 5-Mbits/s token-ring network, calculate the minimum *propagation distance*, d_p, needed for the ring circumference. Assume that the propagation velocity v_p is 200 m/μs.

Solution

$$R = 5 \text{ Mbits/s}$$

Time to emit one bit, t_b:

$$t_b = \frac{1}{5 \times 10^6} \text{ s}$$

Time to emit the 8-bit token, t_t:

$$t_t = \frac{8}{5 \times 10^6} \text{ s}$$

Propagation distance for the 8-bit token:

$$\begin{aligned} d_p &= t_t \times v_p \\ &= \tfrac{8}{5} \text{ μs} \times 200 \text{ m/μs} \\ &= 320 \text{ m} \end{aligned}$$

9.5.3 Performance Comparison of CSMA/CD and Token-Ring Networks

Figure 9.43 compares the delay-throughput characteristics of a CSMA/CD network with a token-ring network. In each case, the cable length is 2 km, there are 50 stations on the network, the average packet length is 1000 bits, and the header size is 24 bits. Figure 9.43a, the case where the transmission rate is 1 Mbits/s, illustrates that under these assumptions, CSMA/CD and token ring perform almost equally well. In Figure 9.43b, only one parameter has been changed as compared to Figure 9.43a; the transmission rate was increased to 10 Mbits/s. The difference for CSMA/CD is considerable; for normalized throughput, $\rho < 0.22$, CSMA/CD performs better than token ring. However, for $\rho > 0.22$, token ring clearly manifests better delay-throughput characteristics. To understand the reason for the poor CSMA/CD performance in Figure 9.43b, let us review the definition of ρ, described in Equations (9.17) and (9.19) and shown as

$$\rho = \frac{b\lambda}{R} = \frac{\rho'}{R}$$

where $\rho' = b\lambda$ is channel throughput in bits per second and R is the channel capacity (maximum transmission bit rate). As R increases, channel throughput must increase accordingly for a given value of ρ. At higher channel throughput rates a significant portion of the CSMA/CD transmission attempts ends in collision [26].

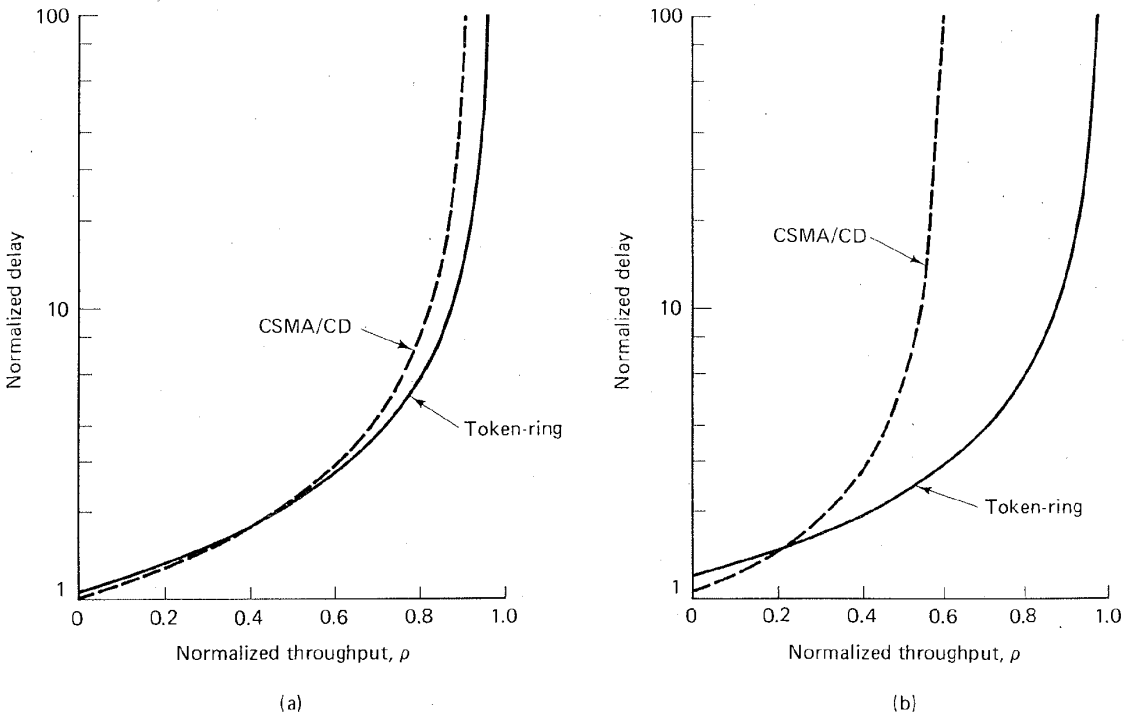


Figure 9.43 Delay versus throughput performance for CSMA/CD and token-ring networks. (a) Transmission rate = 1 Mbits/s. (b) Transmission rate = 10 Mbits/s. (Reprinted with permission from W. Bux, "Local-Area Subnetworks: A Performance Comparison," *IEEE Trans. Commun.*, vol. COM29, no. 10, Oct. 1981, pp. 1465–1473. © 1981 IEEE.)

9.6 CONCLUSION

In this chapter we have outlined the concepts of resource sharing. The classical approaches of FDM/FDMA and TDM/TDMA were discussed in some detail. We also described a hybrid multiple access technique called CDMA, and introduced some of the satellite multiple access techniques that became popular in the 1970s and 1980s, known as multiple-beam frequency reuse and dual-polarization frequency reuse.

We described the demand-assignment (DAMA) techniques in the context of several versions of the ALOHA algorithm, and we considered several of the multiple-access techniques employed with INTELSAT, such as FDM/FM, SPADE, TDMA, and SS/TDMA. Finally, we examined two popular algorithms used for local area networks: carrier-sense multiple access with collision detection (CSMA/CD) and a token-ring network. The goals of the chapter were to introduce an assortment of multiple access techniques rather than attempting a rigorous treatment of any of them.

REFERENCES

1. Rubin, I., "Message Delays in FDMA and TDMA Communication Channels," *IEEE Trans. Commun.*, vol. COM27, no. 5, May 1979, pp. 769–777.
2. Nirenberg, L. M., and Rubin, I., "Multiple Access System Engineering—A Tutorial," *IEEE WESCON/78 Professional Program,* Modern Communication Techniques and Applications, session 21, Los Angeles, Sept. 13, 1978.
3. Abramson, N., "The ALOHA System—Another Alternative for Computer Communications," *Proc. Fall Joint Comput. Conf. AFIPS,* vol. 37, 1970, pp. 281–285.
4. Hayes, J. F., "Local Distribution in Computer Communications," *IEEE Commun. Mag.,* Mar. 1981, pp. 6–14.
5. Schwartz, M., *Computer-Communication Network Design and Analysis,* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1977.
6. Tanenbaum, A. S., *Computer Networks,* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1981.
7. Abramson, N., "The ALOHA System," in N. Abramson and F. F. Kuo, eds., *Computer Communication Networks,* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1973.
8. Kleinrock, L., *Queueing Systems,* Vol. 1, *Theory,* John Wiley & Sons, Inc., New York, 1975.
9. Abramson, N., "Packet Switching with Satellites," *AFIPS Conf. Proc.,* vol. 42, June 1973, pp. 695–702.
10. Rosner, R. D., *Packet Switching,* Lifelong Learning Publications, Wadsworth Publishing Company, Inc., Belmont, Calif., 1982.
11. Crowther, W., Rettberg, R., Walden, D., Ornstein, S., and Heart, F., "A System for Broadcast Communication: Reservation ALOHA," *Proc. Sixth Hawaii Int. Conf. Syst. Sci.,* Jan. 1973, pp. 371–374.
12. Roberts, L., "Dynamic Allocation of Satellite Capacity through Packet Reservation," *AFIPS Conf. Proc.,* vol. 42, June 1973, p. 711.
13. Binder, R., "A Dynamic Packet-Switching System for Satellite Broadcast Channels," *Proc. Int. Conf. Commun.,* June 1975, pp. 41-1–41-5.
14. Capetanakis, J., "Tree Algorithms for Packet Broadcast Channels," *IEEE Trans. Inf. Theory,* vol. IT25, Sept. 1979, pp. 505–515.
15. Puente, J. G., and Werth, A. M., "Demand-Assigned Service for the INTELSAT Global Network," *IEEE Spectrum,* Jan. 1971, pp. 59–69.
16. Jones, J. J., "Hard Limiting of Two Signals in Random Noise," *IEEE Trans. Inf. Theory,* vol. IT9, Jan. 1963, pp. 34–42.
17. Bond, F. E., and Meyer, H. F., "Intermodulation Effects in Limiter Amplifier Repeaters," *IEEE Trans. Commun. Technol.,* vol. COM18, no. 2, Apr. 1970, pp. 127–135.
18. Shimbo, O., "Effects of Intermodulation, AM-PM Conversion, and Additive Noise in Multicarrier TWT Systems," *Proc. IEEE,* vol. 59, Feb. 1971, pp. 230–238.
19. Chakraborty, D. "INTELSAT IV Satellite System (Voice) Channel Capacity versus Earth-Station Performance," *IEEE Trans. Commun. Technol.,* vol. COM19, no. 3, June 1971, 355–362.
20. Campanella, S., and Schaefer, D., "Time Division Multiple Access Systems

(TDMA)," in K. Feher, *Digital Communications, Satellite/Earth Station Engineering,* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1983.

21. Scarcella, T., and Abbott, R. V., "Orbital Efficiency Through Satellite Digital Switching," *IEEE Commun. Mag.,* May 1983, pp. 38–46.
22. Muratani, T., Satellite-Switched Time-Domain Multiple Access," *Proc. IEEE Electron. and Aerosp. Conf. (EASCON),* 1974, pp. 189–196.
23. Dill, G. D., "TDMA, The State-of-the-Art," *Rec. IEEE Electron. Aerosp. Syst. Conv. (EASCON),* Sept. 26–28, 1977, pp. 31-5A–31-5I.
24. Jarett, K., "Operational Aspects of Intelsat VI Satellite-Switched TDMA Communication System," *AIAA Tenth Commun. Satell. Syst. Conf.* Mar. 1984, pp. 107–111.
25. Stallings, W., "Local Network Performance," *IEEE Commun. Mag.,* vol. 22, No. 2, Feb. 1984, pp. 27–36.
26. Bux, W., Local-Area Subnetworks: A Performance Comparison," *IEEE Trans. Commun.,* vol. COM29, no. 10, Oct. 1981, pp. 1465–1473.
27. Dixon, R. C., Strole, N. C., and Markov, J. D., "A Token-Ring Network for Local Data Communications," *IBM Syst. J.,* vol. 22, no. 1–2, 1983, pp. 47–62.

PROBLEMS

9.1. Design an FDM signal set consisting of five voice channels, each in the frequency range 300 to 3400 Hz. The multiplexed composite is to be made up of inverted sidebands and is to occupy the spectral region from 30 to 50 kHz.
- **(a)** Draw the composite spectrum, indicating individual spectrum and guard band frequency locations.
- **(b)** Draw a block diagram showing the heterodyning and filtering details and the required local oscillator values.

9.2. A recciver is tuned to receive the lower sideband (LSB) of a radio-frequency (RF) carrier wave with frequency, $f_c = 8$ MHz. The bandwidth of the LSB signal is 100 kHz. The receiver employs a local oscillator (LO) with frequency, f_{LO}, for heterodyning the received signal down to a lower intermediate frequency (IF). Assume that $f_{LO} > f_c$, and that the IF amplifier is centered at 2 MHz. Draw a block diagram of the heterodyning conversion, including the RF filter, the LO, and the IF filter. Indicate the center frequency of each filter and typical spectra of the signals at various points in the diagram.

9.3. Equations (9.13) to (9.15) demonstrate that the average message delay time for TDMA is less than that for FDMA. Discuss the practical benefits of such reduced delay in TDMA, as a function of frame time, for a satellite link with a one-way range of 36,000 km. For what values of frame time can there be a significant advantage of TDMA over FDMA?

9.4. A group of stations share a 56-kbits/s pure ALOHA channel. Each station outputs a packet on the average of once every 10 s, even if the previous one has not yet been sent (i.e., the stations buffer the packets). Each packet is comprised of 3000 bits. What is the maximum number of stations that can share this channel, assuming that the arrival process is Poisson?

9.5. A group of three stations share a 56-kbits/s pure ALOHA channel. The average bit

rate transmitted from each of the three stations is $R_1 = 7.5$ kbits/s, $R_2 = 10$ kbits/s, and $R_3 = 20$ kbits/s. The size of each packet is 100 bits/packet. Find the normalized total traffic on the channel, the normalized throughput, the probability of successful transmission, and the arrival rate of successful packets. Assume that the arrival process is Poisson.

9.6. Verify that for a pure ALOHA access scheme, the normalized throughput is bounded by $1/2e$ and that this maximum occurs when the normalized total traffic is equal to 0.5.

9.7. **(a)** Verify that Equation (9.24) is a valid probability density function (pdf) for a discrete random variable.

(b) Calculate the mean of a discrete random variable having a pdf like the one given in Equation (9.24).

(c) Show that your result in part (b) is consistent with the claim that λ is the average packet arrival rate.

9.8. Consider the pure ALOHA arrival scenario shown in Figure P9.1. The vertical arrows indicate packet arrival times. N_n is the number of arriving packets in the time interval $(T_{n-1}, T_n]$, where $(t_x, t_y]$ indicates the interval $t_x < t \leq t_y$. N_{n+1} is the number of arriving packets in $(T_n, T_{n+1}]$, and τ is the time duration per packet in seconds. The average arrival rate is λ_t. Assume the arrivals are independent of each other.

(a) Write an expression for the joint pdf of N_n and N_{n+1}.

(b) Let T_n define the time at which user A's packet arrives. Express, in terms of the joint pdf of N_n and N_{n+1}, the probability that user A's transmission will be successful.

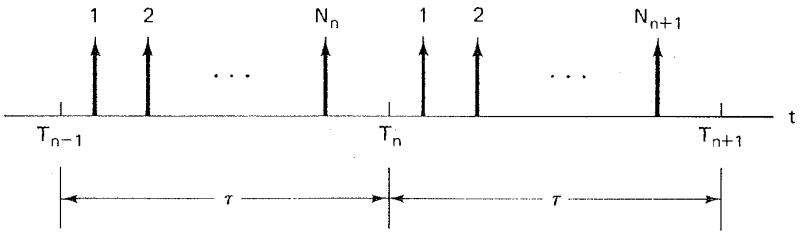


Figure P9.1

9.9. Let $N = N_n + N_{n+1}$, where N_n and N_{n+1} are as defined in Problem 9.8. Write an expression for the pdf of N, and give an interpretation for N.

9.10. Six thousand stations are competing for the use of a single slotted ALOHA channel. The average station makes 30 requests per hour, where each request is for one slot of 500-μs duration. Calculate the normalized total traffic on the channel.

9.11. Consider the arrival scenario of Figure P9.1; the location of the packet arrival times are permissible as shown under pure ALOHA, but not under slotted ALOHA, where arrivals are permitted only at the discrete times T_i, where $i = 0, 1, \ldots$. Assume that the average arrival rate is λ_t.

(a) How would Figure P9.1 need to be modified if slotted ALOHA is used? How would the pdfs of N_n and N_{n+1} change?

(b) If user A's packet arrives at time T_n, what is the probability of successful transmission?

9.12. A group of slotted-ALOHA stations generate a total of 120 requests per second,

including both original and retransmissions. Each request is for a 12.5-ms duration slot.

(a) What is the normalized total traffic on the channel?
(b) What is the probability of a successful transmission on the first attempt?
(c) What is the probability of exactly two collisions before a successful transmission?

9.13. Measurements of a slotted-ALOHA channel show that 20% of the slots are idle.
(a) What is the normalized total traffic on the channel?
(b) What is the normalized throughput?
(c) Is the channel underloaded or overloaded?

9.14. Show that the sum of two Poisson processes, with rates λ_1 and λ_2, is also a Poisson process, with rate $\lambda_t = \lambda_1 + \lambda_2$. Generalize your result for the sum of n Poisson processes.

9.15. A 10-MHz transponder is occupied by 200 identical carriers, half servicing stations with $G/T = 40$ dB/K, the other half servicing stations with $G/T = 37$ dB/K. All stations have a requirement to operate with a bit error probability of 10^{-5}. The transponder is power limited under this configuration.
(a) What is the maximum possible bandwidth for each carrier?
(b) Suppose that each carrier has a bandwidth of 40 kHz, and the transponder is required to service a group of larger ($G/T = 40$ dB/K) stations only. How many stations can the transponder handle? Will the transponder be power or bandwidth limited?
(c) Repeat part (b) for the case where the transponder is to service a group of small ($G/T = 37$ dB/K) stations only.

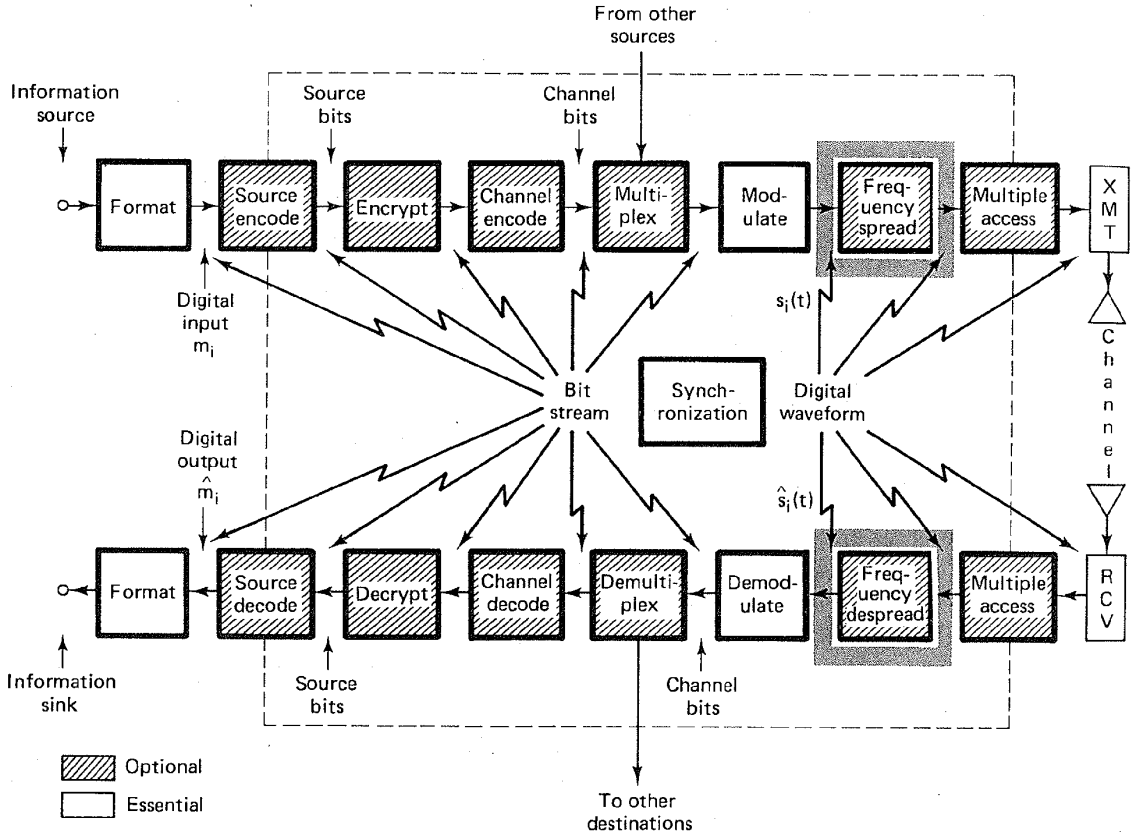
9.16. A TDMA system operates at 100 Mbits/s with a 2-ms frame time. Assume that all slots are of equal length and that a guard time of 1 μs is required between slots.
(a) Compute the efficiency of the communications resource (CR) for the case of 1, 2, 5, 10, 20, 50, and 100 slots per frame.
(b) Repeat part (a) assuming that a 100-bit preamble is required at the start of each slot. Compute the efficiency of the CR in terms of the desired information transmission.
(c) Graph the results of parts (a) and (b).

9.17. With reference to Equation (9.36):
(a) Discuss the efficiency of the CR use if all S_i and R_j are equal.
(b) Discuss the effect of a few S_i or R_j being much larger than the majority. How can the efficiency of the CR be improved?
(c) When are the distributions of S_i and R_j likely to be similar? Dissimilar?

9.18. (a) Consider a token-ring network operating at a transmission rate of 10 Mbits/s over a cable having a propagation velocity of 200 m/μs. How many meters of cable is equal to a delay of 1 bit at each ring interface?
(b) If the token is 10 bits long, and all but three stations are switched off during evening hours, what is the minimum cable length needed for the ring?

CHAPTER 10

Spread-Spectrum Techniques



10.1 SPREAD-SPECTRUM OVERVIEW

The initial application of spread-spectrum (SS) techniques was in the development of military guidance and communication systems. By the end of World War II, spectrum spreading for jamming resistance was already a familiar concept to radar engineers [1], and during subsequent years, SS investigation was motivated primarily by the desire to achieve highly jam-resistant communication systems. As a result of this research, there emerged an assortment of other applications in such areas as energy density reduction, high-resolution ranging, and multiple access, which will be discussed in later sections. The techniques considered in this chapter are called *spread spectrum* because the transmission bandwidth employed is much greater than the minimum bandwidth required to transmit the information. A system is defined to be a spread-spectrum system if it fulfills the following requirements:

1. The signal occupies a bandwidth much in excess of the minimum bandwidth necessary to send the information.
2. Spreading is accomplished by means of a *spreading signal*, often called a *code signal*, which is independent of the data. The details of some spreading signals are described in later sections.
3. At the receiver, despreading (recovering the original data) is accomplished by the correlation of the received spread signal with a synchronized replica of the spreading signal used to spread the information.

Standard modulation schemes such as frequency modulation and pulse code modulation also spread the spectrum of an information signal, but they do not qualify as spread-spectrum systems since they do not satisfy all the conditions outlined above.

10.1.1 The Beneficial Attributes of Spread-Spectrum Systems

10.1.1.1 Interference Suppression Benefits

White Gaussian noise is a mathematical model that, by definition, has infinite power spread uniformly over all frequencies. Effective communication is possible with this interfering noise of infinite power because only the finite-power noise components that are present within the signal space (in other words, share the *same coordinates* as the signal components) can interfere with the signal. The balance of the noise power may be thought of as noise that is effectively tuned out by the detector (see Section 3.2.2). For a typical narrowband signal, this means that only the noise in the signal bandwidth can degrade performance. The idea behind a spread-spectrum anti-jam (AJ) system is as follows. Consider that many orthogonal signal coordinates or dimensions are available to a communication link and that only a small subset of these signal coordinates are used at any time. We assume that the jammer cannot determine the signal subset that is currently in use. For signals of bandwidth W and duration T, the number of signaling dimensions can be shown [2] to be approximately $2WT$. Given a specific signal design, the error performance of such a system is only a function of E_b/N_0. Against white Gaussian noise, with *infinite* power, the use of spreading (large $2WT$) offers no performance improvement. However, when the noise stems from a jammer with a *fixed finite* power and with uncertainty as to where in the signal space the signal coordinates are located, the jammer's choices are limited to those shown below.

1. Jam *all* the signal coordinates of the system, with an *equal* amount of power in each one, with the result that *little* power is available for each coordinate.
2. Jam a *few* signal coordinates with *increased* power in each of the jammed coordinates (or more generally, jam all the coordinates with various amounts of power in each).

Figure 10.1 compares the effect of spectrum spreading in the presence of white noise with spreading in the presence of an intentional jammer. The power spectral density of the signal is denoted $G(f)$ before spreading, and $G_{ss}(f)$ after spreading. For simplicity, the figure treats the frequency dimension only. In Figure 10.1a it can be seen that the single-sided power spectral density of white noise, N_0, is unchanged as a result of expanding the signal bandwidth from W to W_{ss}. The average power of white noise (area under the spectral density curve) is infinite. Hence, the use of spreading offers no performance improvement here. Figure 10.1b (upper diagram) illustrates the case of received (fixed finite) jammer power, J, and power spectral density, $J_0' = J/W$, where W is the unspread band-

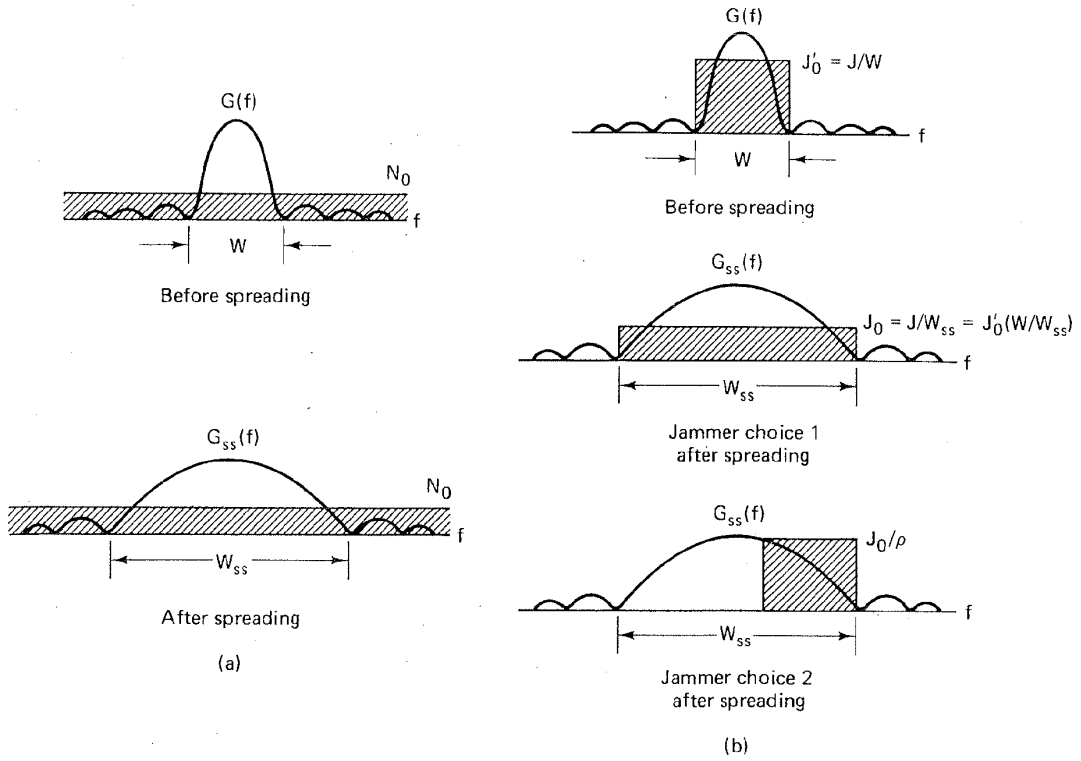


Figure 10.1 Effect of spectrum spreading. (a) Spectrum spreading in the presence of white noise. (b) Spectrum spreading in the presence of an intentional jammer.

width being jammed. Once the signal bandwidth is spread, the jammer can make one of the two choices listed earlier—choice 1 results in a reduction in jammer noise spectral density, J_0', by a factor (W/W_{ss}) across the spread spectrum. The resulting noise spectral density, $J_0 = J/W_{ss}$, is referred to as the *broadband jammer noise spectral density*. Choice 2 results in a reduction in the number of signal coordinates that the jammer occupies. However, with choice 2 the jammer can increase its noise spectral density from J_0 to J_0/ρ $(0 < \rho \leq 1)$, where ρ is the portion of the spread-spectrum band the jammer elects to jam. If the jammer makes a poor choice in the coordinates to be jammed, the average effect of jamming will be less than if it makes a good choice. The larger the dimensionality of the signal set or the more signal coordinates the communicator can choose from, the greater is the jammer's uncertainty regarding the effectiveness of the jamming technique, and the better will be the protection against jamming.

Jamming is not always the result of an intentional act. Sometimes, the jamming signal is caused by natural phenomena, and sometimes it is the result of self-interference caused by *multipath*, in which delayed versions of the signal, arriving via alternative paths, interfere with the direct path transmission.

10.1.1.2 Energy Density Reduction

One can imagine situations where it is desired that a communications link be operated without being detected by anyone other than the intended receiver. Systems designed for this special task are known as *low probability of detection* (LPD) or *low probability of intercept* (LPI) communication systems. These systems are designed to make the detection of their signals as difficult as possible by anyone but the intended receiver. The goal of such a system is to use the minimum signal power and the optimum signaling scheme that results in the minimum probability of being detected. Since, in spread-spectrum systems, the signal is spread over many more signaling coordinates than in conventional modulation schemes, the resulting signal power is, on the average, spread thinly and uniformly in the spread domain. Therefore, not only can the spread-spectrum signal be made difficult to jam, but additionally, the signal's very existence may be rendered difficult to perceive. To anyone who does not possess a synchronized replica of the spreading signal, the spread-spectrum signal will seem "buried in the noise."

A *radiometer* is a simple power measuring instrument that can be used by an adversary to detect the presence of spread-spectrum signals within some bandwidth W. The radiometer, illustrated in Figure 10.2, consists of a bandpass filter (BPF) with bandwidth W, a squaring circuit to ensure a positive output value, since the presence of *signal energy* is being detected, and an integrating circuit. At time $t = T$, the output of the integrator is compared to a preset threshold. If the output of the integrator is larger than the threshold, a signal is declared present; otherwise, the signal is declared absent. References [3, 4] provide details on the detectability of spread-spectrum signals, using radiometers and other more complicated instruments that make use of the features of the SS signal itself.

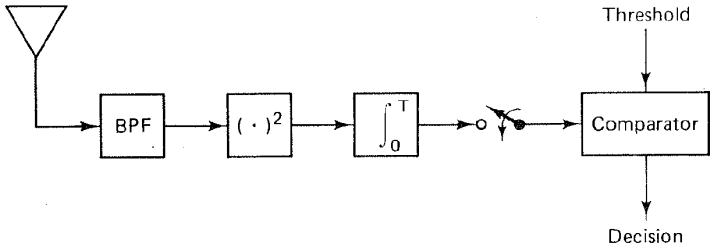


Figure 10.2 Radiometer.

Spread-spectrum systems that are designed to exhibit LPI may also exhibit a *low probability of position fix* (LPPF), which means that even if the presence of the signal is perceived, the direction of the transmitter is difficult to pinpoint. Some spread-spectrum systems also exhibit a *low probability of signal exploitation* (LPSE), which means that the identification of the source is difficult to ascertain.

Another, unrelated application of spread-spectrum signaling deals with the fact that in some cases energy density reduction may be required to meet national allocation regulations. Downlink transmissions from satellites must meet international regulations on the spectral density that impinges on the earth. By spread-

ing the downlink energy over a wider bandwidth, the total transmitted power can be increased and hence performance improved, while the energy density regulations are followed.

10.1.1.3 Fine Time Resolution

Spread-spectrum signals can be used for ranging or determination of position location. Distance can be determined by measuring the time delay of a pulse as it traverses the channel. Uncertainty in the delay measurement is inversely proportional to the bandwidth of the signal pulse. This can be seen by the illustration in Figure 10.3. The uncertainty of the measurement, Δt, is proportional to the rise time of the pulse, which is inversely proportional to the bandwidth of the pulse signal; that is,

$$\Delta t \simeq \frac{1}{W} \tag{10.1}$$

The larger the bandwidth, the more precisely one can measure range. Over a Gaussian channel, a one-shot measurement on a single pulse is not very reliable. The spread-spectrum technique, however, uses a code signal consisting of a long sequence of polarity changes (e.g., a binary PSK-modulated signal) in place of the single pulse. Upon reception, the received sequence is correlated against a local replica and the results of the correlation are used to perform an accurate time-delay or range measurement.

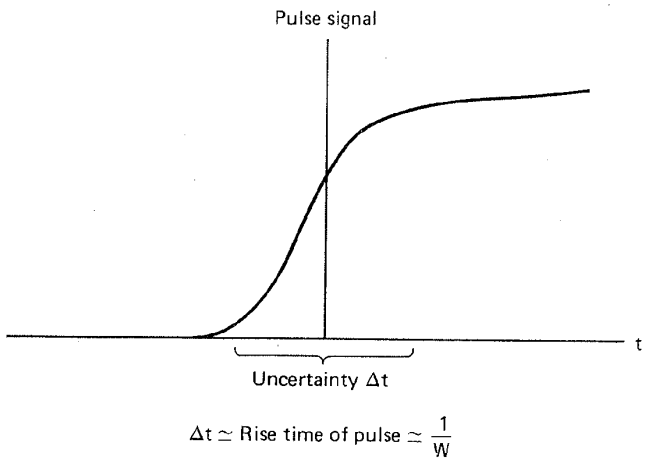


Figure 10.3 Time-delay measurement.

10.1.1.4 Multiple Access

Spread-spectrum methods can be used as a multiple access technique, in order to share a communications resource among numerous users in a coordinated manner. The technique, termed *code-division multiple access* (CDMA), since each

simultaneous user employs a unique spread-spectrum signaling code, was discussed briefly in Chapter 9. One of the by-products of this type of multiple access is the ability to provide communication privacy between users with different spreading signals. An unauthorized user (a user not having access to a spreading signal) cannot easily monitor the communications of the authorized users. A more detailed treatment is presented in a later section.

10.1.2 Model for Spread-Spectrum Interference Rejection

Figure 10.4 illustrates a model for spread-spectrum interference rejection. At the modulator, the information signal $x(t)$, with a data rate of R bits/s, is multiplied by a spreading code signal; $g(t)$, having a code symbol rate, usually called the code *chip rate*, R_p chips/second. Assume that the transmission bandwidths for $x(t)$ and $g(t)$ are R hertz and R_p hertz, respectively. Multiplication in the time domain transforms to convolution in the frequency domain:

$$x(t)g(t) \leftrightarrow X(\omega) * G(\omega) \tag{10.2}$$

Therefore, if the data signal is narrowband compared to the spreading signal, the resulting product signal $x(t)g(t)$ will have approximately the bandwidth of the spreading signal (see Section A.5).

At the demodulator, the received signal is ideally multiplied by a synchronized replica of the spreading code signal, $g(t)$, which results in the despreading of the signal. A filter with bandwidth R is used to remove any spurious higher-frequency components. If there is any undesired signal at the receiver, the multiplication by $g(t)$ will spread this undesired signal, in the same way that the multiplication by $g(t)$ at the transmitter spread the desired signal originally. Consider the effect on a jammer that attempts to position a narrowband jamming signal within the information bandwidth. The first operation at the receiver input is multiplication by the spreading signal. Hence the jamming tone is spread to the bandwidth of the spreading signal.

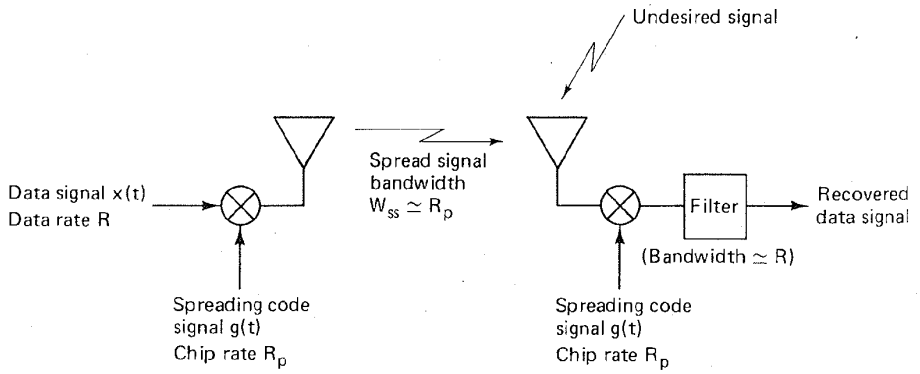


Figure 10.4 Basic spread-spectrum technique.

The essence behind the interference rejection capability of a spread-spectrum system can be summarized as follows:

1. Multiplication by the spreading signal *once* spreads the signal bandwidth.
2. Multiplication by the spreading signal *twice*, followed by filtering, recovers the original signal.
3. The desired signal gets multiplied *twice*, but the interference signal gets multiplied only *once*.

10.1.3 A Catalog of Spreading Techniques

Figure 10.5 highlights the popular techniques for spreading the information signal over a large number of signal coordinates or dimensions. For signals of bandwidth W and duration T, the dimensionality of the signaling space is approximately $2WT$.

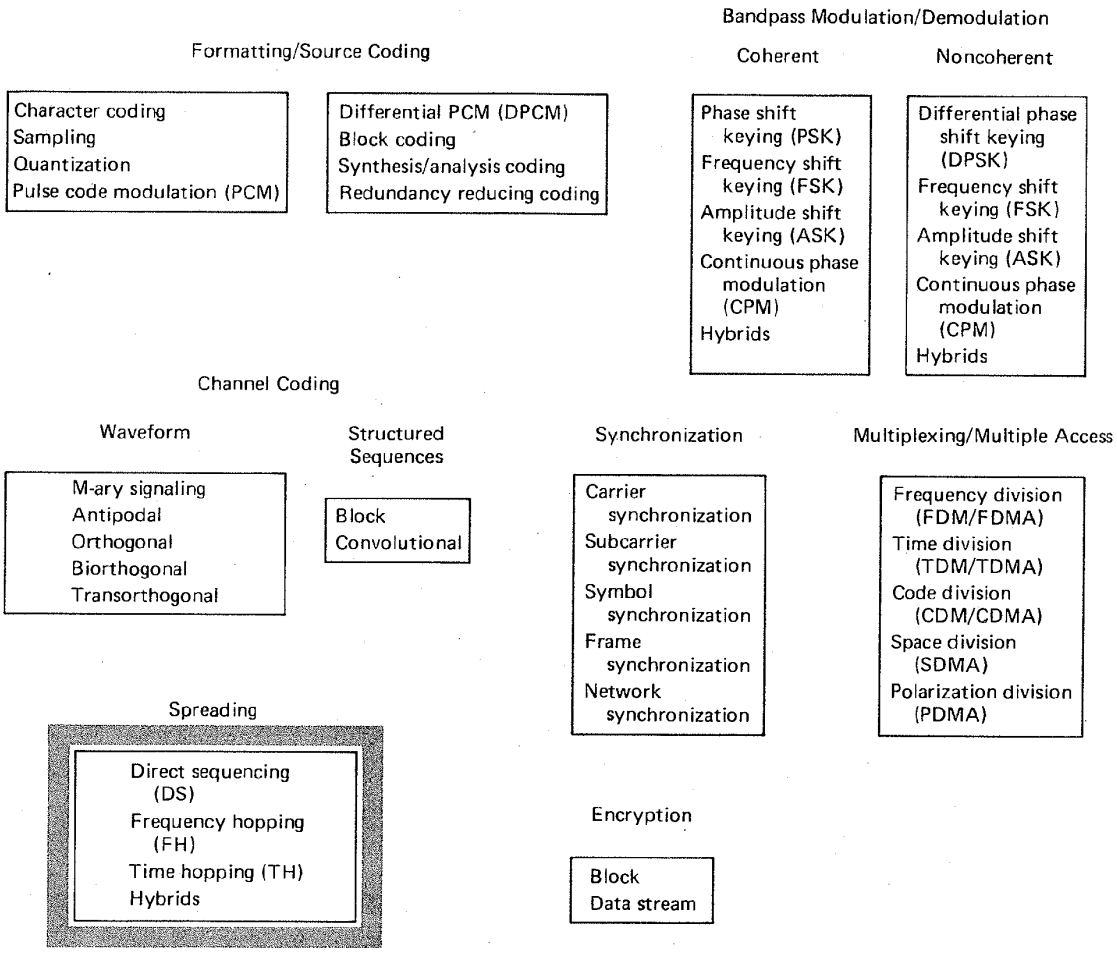


Figure 10.5 Basic digital communication transformations.

To increase the dimensionality, we can either increase W by spectrum spreading, or increase T by time spreading or time hopping (TH). With spectrum spreading the signal is spread in the frequency domain. With time hopping, a message with data rate R is allocated a longer transmission-time duration than would be used with a conventional modulation scheme. During this longer time the data are sent in bursts according to the dictates of a code. We can say that with time hopping the signal is spread in the time domain. For both cases, frequency spreading and time spreading, a jammer will be uncertain regarding the signaling subset that is currently in use.

In Figure 10.5, the first two items listed under the category of spreading, *direct sequencing* (DS) and *frequency hopping* (FH), are the most commonly used techniques for spectrum spreading. As a jamming-rejection technique, *time hopping* (TH), the third item in the list, is similar to spread spectrum, in that the location of the signal coordinates is hidden from potential adversaries. Also, there are hybrid combinations of the spreading techniques, for example, DS/FH, FH/TH, and DS/FH/TH; however, these techniques can be viewed as simple extensions of the material presented here and will not be elaborated on. In this chapter we focus only on the two major spread-spectrum techniques, direct sequencing and frequency hopping.

10.1.4 Historical Background

10.1.4.1 Transmitted Reference versus Stored Reference

During the early years of spread-spectrum investigation one technique that was considered for operating a transmitter and receiver synchronously with a *truly random* spreading signal such as wideband noise, was called a *transmitted reference* (TR) system. In a TR system the transmitter would send two versions of an unpredictable wideband carrier, one modulated by data and the other unmodulated. These two signals were transmitted on separate channels. The receiver used the unmodulated carrier as the reference signal for despreading (correlating) the data-modulated carrier. The principal advantage of a TR system was that there were no significant synchronization problems at the receiver, since the data-modulated signal and the spreading signal used for despreading were transmitted simultaneously. The principal disadvantages of TR systems were that (1) the spreading code was sent in the clear and thus was available to any listener; (2) the system could be easily spoofed by a jammer sending a pair of waveforms acceptable to the receiver; (3) performance degraded at low signal levels since noise was present on both signals; and (4) twice the bandwidth and transmitted power were required because of the need to transmit the reference.

Modern spread-spectrum systems all use a technique called *stored reference* (SR), whereby the spreading code signal is independently generated at both the transmitter and the receiver. The main advantage of an SR system is that a well-designed code signal cannot be predicted by monitoring the transmission. Note that the noiselike code signal in an SR system cannot be truly random as it could in the case of a TR system. Since the same code must be generated independently

at two or more sites, the code sequence must be deterministic, even though it should appear random to unauthorized listeners. Such random-appearing deterministic signals are called pseudonoise (PN) or pseudorandom signals; their generation is treated later in greater detail.

10.1.4.2 Noise Wheels

In the late 1940s and early 1950s, Mortimer Rogoff, working at ITT, demonstrated the fundamental operation of spectrum spreading systems with a novel experiment [5]. Using photographic techniques, Rogoff built a "noise wheel" for storing a noiselike signal. He randomly selected 1440 numbers not ending in 00 from the Manhattan telephone directory, and radially plotted the middle two of the last four digits so that the radius at every $\frac{1}{4}°$ represented a new random number. The drawing was transferred to the wheel-shaped film shown in Figure 10.6. When the wheel was rotated past a slit of light, the resulting intensity-modulated light beam provided a stored noiselike spreading signal to be sensed by a photocell.

Rogoff mounted two such identical wheels on a single axis driven by a 900-rpm synchronous motor. One wheel's noiselike spreading signal was modulated

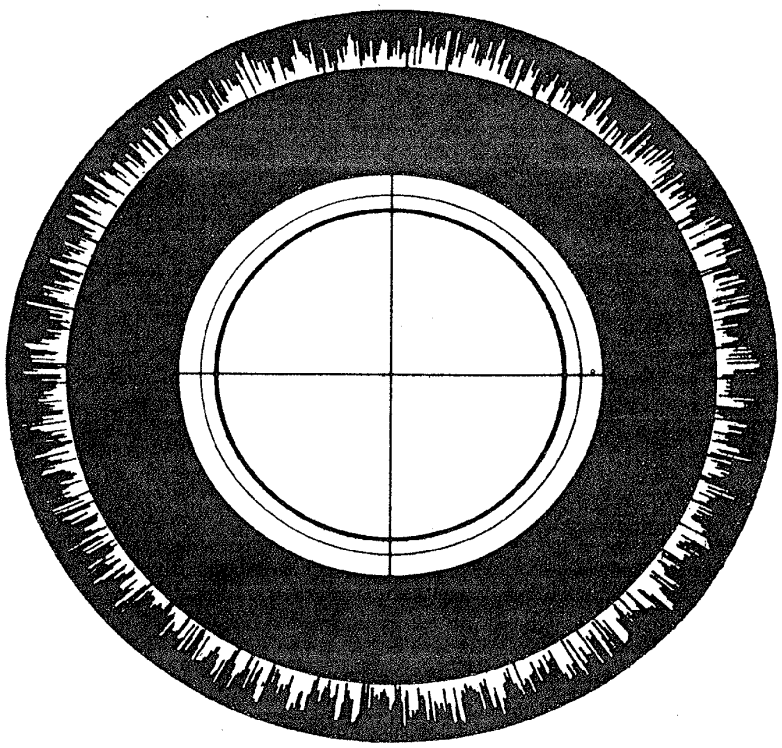


Figure 10.6 Rogoff's noise wheel. [Reprinted from Section I (Communications) of "Application of Statistical Methods to Secrecy Communication Systems," Proposal 946, Fed. Telecomm. Lab., August 28, 1950, Fig. 6, courtesy of ITT.]

with data (and interference) to provide one input to the receiving correlator, while the other wheel's unmodulated spreading signal provided the other input to the corrclator. These baseband experiments, performed with data rates of 1 bit/s, demonstrated the feasibility of conveying information hidden in noiselike signals [6].

10.2 PSEUDONOISE SEQUENCES

The spread-spectrum approach called *transmitted reference* (TR) can utilize a *truly* random code signal for spreading and despreading, since the code signal and the data-modulated code signal are simultaneously transmitted over different regions of the spectrum. The *stored reference* (SR) approach *cannot* use a truly random code signal since the code needs to be stored or generated at the receiver. For the SR system a *pseudonoise* or *pseudorandom* code signal must be used.

How does a pseudorandom signal differ from a random one? A random signal *cannot* be predicted; its future variations can only be described in a statistical sense. However, a pseudorandom signal is not random at all; it is a deterministic, periodic signal that is known to both the transmitter and receiver. Why the name "pseudonoise" or "pseudorandom"? Even though the signal is deterministic, it appears to have the statistical properties of sampled white noise. It appears, to an unauthorized listener, to be a truly random signal.

10.2.1 Randomness Properties

What are these randomness properties that make a pseudorandom signal appear truly random? There are three basic properties that can be applied to any periodic binary sequence as a test for the appearance of randomness. The properties, called *balance, run,* and *correlation*, are described below for binary signals:

1. *Balance property*. Good balance requires that in each period of the sequence, the number of binary ones differs from the number of binary zeros by at most one digit.
2. *Run property*. A *run* is defined as a sequence of a single type of binary digit(s). The appearance of the alternate digit in a sequence starts a new run. The length of the run is the number of digits in the run. Among the runs of ones and zeros in each period, it is desirable that about one-half the runs of each type are of length 1, about one-fourth are of length 2, one-eighth are of length 3, and so on.
3. *Correlation property*. If a period of the sequence is compared term by term with any cyclic shift of itself, it is best if the number of agreements differs from the number of disagreements by not more than one count.

In the next section, a PN sequence is generated to test these properties.

10.2.2 Shift Register Sequences

Consider the linear feedback shift register illustrated in Figure 10.7. It is made up of a four-stage register for storage and shifting, a modulo-2 adder, and a feedback path from the adder to the input of the register (modulo-2 addition has been defined in Section 2.12.3). The shift register operation is controlled by a sequence of clock pulses (not shown). At each clock pulse the contents of each stage in the register is shifted one stage to the right. Also, at each clock pulse the contents of stages X_3 and X_4 are modulo-2 added (a linear operation), and the result is fed back to stage X_1. The shift register sequence is defined to be the output of the last stage—stage X_4 in this example.

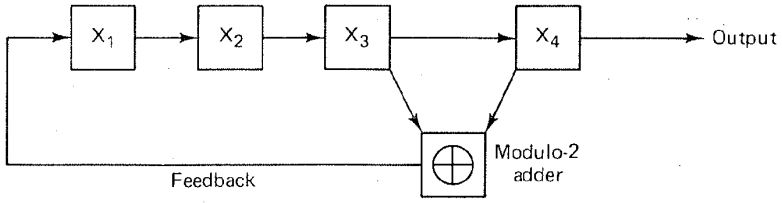


Figure 10.7 Linear feedback shift register example.

Assume that stage X_1 is initially filled with a one and the remaining stages are filled with zeros, that is, the initial state of the register is 1 0 0 0. From Figure 10.7 we can see that the succession of register states will be as follows:

1 0 0 0 0 1 0 0 0 0 1 0 1 0 0 1 1 1 0 0 0 1 1 0 1 0 1 1 0 1 0 1

1 0 1 0 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 0 0 1 1 0 0 0 1 1 0 0 0

Since the last state, 1 0 0 0, corresponds to the initial state, we see that the register repeats the foregoing sequence after 15 clock pulses. The output sequence is obtained by noting the contents of stage X_4 at each clock pulse. The output sequence is seen to be

0 0 0 1 0 0 1 1 0 1 0 1 1 1 1

where the leftmost bit is the earliest bit. Let us test the sequence above for the randomness properties outlined in the preceding section. First, the balance property; there are seven zeros and eight ones in the sequence—therefore, the sequence meets the balance condition. Next, the run property; consider the zero runs—there are four of them. One-half are of length 1, and one-fourth are of length 2. The same is true for the one runs. The sequence is too short to go further, but we can see that the run condition is met. The correlation property is treated in Section 10.2.3.

The shift register generator produces sequences that depend on the number of stages, the feedback tap connections, and initial conditions. The output sequences can be classified as either *maximal length* or *nonmaximal length*. Max-

imal length sequences have the property that for an n-stage linear feedback shift register the sequence repetition period in clock pulses p is

$$p = 2^n - 1 \tag{10.3}$$

Thus it can be seen that the sequence generated by the shift register generator of Figure 10.7 is an example of a maximal length sequence. If the sequence length is less than $(2^n - 1)$, the sequence is classified as a nonmaximal length sequence.

10.2.3 PN Autocorrelation Function

The autocorrelation function $R_x(\tau)$ of a periodic waveform $x(t)$, with period T_0, was given in Equation (1.23) and is shown below in normalized form.

$$R_x(\tau) = \frac{1}{K}\left(\frac{1}{T_0}\right)\int_{-T_0/2}^{T_0/2} x(t)x(t+\tau)\,dt \qquad \text{for } -\infty < \tau < \infty \tag{10.4}$$

where

$$K = \frac{1}{T_0}\int_{-T_0/2}^{T_0/2} x^2(t)\,dt \tag{10.5}$$

When $x(t)$ is a periodic pulse waveform representing a PN code, we refer to each fundamental pulse as a *PN code symbol* or a *chip*. For such a PN waveform of unit chip duration and period p chips, the normalized autocorrelation function may be expressed as

$$R_x(\tau) = \frac{1}{p} \cdot \begin{pmatrix}\text{number of agreements less number of disagreements}\\ \text{in a comparison of one full period of the sequence}\\ \text{with a } \tau \text{ position cyclic shift of the sequence}\end{pmatrix} \tag{10.6}$$

The normalized autocorrelation function for a maximal length sequence, $R_x(\tau)$, is shown plotted in Figure 10.8. It is clear that for $\tau = 0$, that is, when $x(t)$ and its replica are perfectly matched, $R(\tau) = 1$. However, for any cyclic shift between $x(t)$ and $x(t + \tau)$ with $(1 \leq \tau < p)$, the autocorrelation function is equal to $-1/p$ (for large p, the sequences are virtually decorrelated for a shift of a *single chip*).

It is now easy to test the output PN sequence of the shift register in Figure 10.7 for the third randomness property—correlation. Below is shown the output sequence; also shown is the same sequence with a single end-around shift:

$$\begin{array}{c} 0\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 1 \\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 1 \\ \hline d\ a\ a\ d\ d\ a\ d\ a\ d\ d\ d\ d\ a\ a\ a \end{array}$$

The digits that agree are labeled a and those that disagree are labeled d. Following Equation (10.6), the value of the autocorrelation function for this single one-chip shift is seen to be

$$R(\tau = 1) = \tfrac{1}{15}(7 - 8) = -\tfrac{1}{15}$$

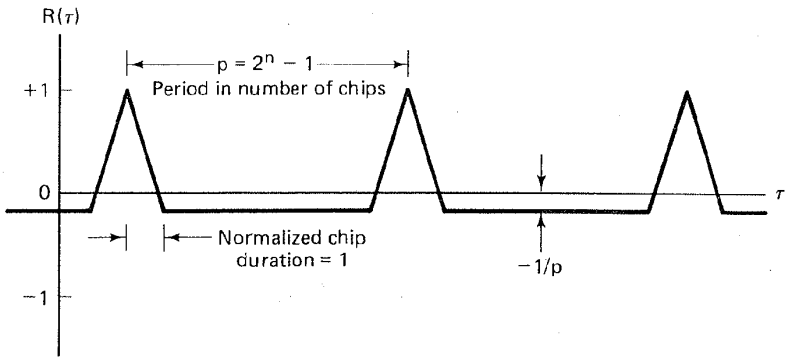


Figure 10.8 PN autocorrelation function.

Any cyclic shift yielding a mismatch from perfect synchronization results in the same autocorrelation value, $-1/p$. Hence the sequence meets the third randomness property.

10.3 DIRECT-SEQUENCE SPREAD-SPECTRUM SYSTEMS

The block diagram in Figure 10.9a depicts a *direct-sequence* (DS) modulator. "Direct sequence" is the name given to the spectrum spreading technique whereby a carrier wave is first modulated with a data signal $x(t)$, then the data-modulated signal is again modulated with a high-speed (wideband) spreading signal $g(t)$. Consider a constant-envelope data-modulated carrier having power P, radian frequency ω_0, and data phase modulation $\theta_x(t)$, given by

$$s_x(t) = \sqrt{2P} \cos [\omega_0 t + \theta_x(t)] \tag{10.7}$$

Upon further constant-envelope modulation by the spreading signal, $g(t)$, the transmitted waveform can be expressed as

$$s(t) = \sqrt{2P} \cos [\omega_0(t) + \theta_x(t) + \theta_g(t)] \tag{10.8}$$

where the phase of the carrier is now seen to have two components: $\theta_x(t)$ due to the data and $\theta_g(t)$ due to the spreading sequence.

In Chapter 3 it was shown that ideal suppressed carrier binary phase shift keying (BPSK) modulation results in instantaneous changes of π radians to the phase of the carrier, according to the dictates of the data. We can equivalently express Equation (10.7) as the multiplication of the carrier wave by $x(t)$, an antipodal pulse stream with pulse values of $+1$ or -1:

$$s_x(t) = \sqrt{2P}\, x(t) \cos \omega_0 t \tag{10.9}$$

If, like the data, the spreading sequence modulation is also BPSK, and $g(t)$ is an antipodal pulse stream with pulse values of $+1$ or -1, Equation (10.8) can be written as

$$s(t) = \sqrt{2P}\, x(t)g(t) \cos \omega_0 t \qquad (10.10)$$

A modulator based on Equation (10.10) is illustrated in Figure 10.9b. The data pulse stream and the spreading pulse stream are first multiplied, and then the composite $x(t)g(t)$ modulates the carrier. If the assignment of pulse value to binary value is

Pulse value	Binary value
1	0
−1	1

then the initial step in the DS/BPSK modulation can be accomplished by the modulo-2 addition of the binary data sequence with the binary spreading sequence.

Demodulation of the DS/BPSK signal is accomplished by correlating or re-modulating the received signal with a synchronized replica of the spreading signal $g(t - \hat{T}_d)$ as seen in Figure 10.9c, where $\hat{T}_d$ is the receiver's estimate of the propagation delay T_d from the transmitter to the receiver. In the absence of noise and interference, the output signal from the correlator can be written as

$$A\sqrt{2P}\, x(t - T_d)g(t - T_d)g(t - \hat{T}_d) \cos [\omega_0(t - T_d) + \phi] \qquad (10.11)$$

where the constant A is a system gain parameter and ϕ is a random phase angle in the range $(0, 2\pi)$. Since $g(t) = \pm 1$, the product $g(t - T_d)g(t - \hat{T}_d)$ will be unity if $\hat{T}_d = T_d$, that is, if the code signal at the receiver is exactly synchronized with the code signal at the transmitter. When it is synchronized, the output of the receiver correlator is the despread data-modulated signal (except for a random phase ϕ and delay T_d). The despreading correlator is then followed by a conventional demodulator for recovering the data.

10.3.1 Example of Direct Sequencing

Figure 10.10 is an example of DS/BPSK modulation and demodulation following the block diagrams of Figure 10.9b and c. In Figure 10.10a are shown the binary data sequence (1, 0) and its bipolar pulse waveform equivalent $x(t)$, where the binary to pulse value assignments are the same as those described in the preceding section. Examples of a binary spreading sequence and its bipolar pulse waveform equivalent $g(t)$ are shown in Figure 10.10b. The modulo-2 addition of the data sequence and the code sequence, and the equivalent waveform of the product $x(t)g(t)$, is shown in Figure 10.10c.

For the BPSK modulation described by Equations (10.8) and (10.10), it is shown in Figure 10.10d that the phase of the carrier, $\theta_x(t) + \theta_g(t)$, equals π when the value of the product waveform $x(t)g(t)$ equals -1 (or the modulo-2 sum of data and code is binary 1). Similarly, the phase of the carrier is zero when the value of $x(t)g(t)$ equals $+1$ (or the modulo-2 sum of data and code is binary 0). One can appreciate the *signal hiding* property of spread-spectrum signals by comparing the code waveform in Figure 10.10b with the composite waveform in Figure 10.10c. The latter has the signal $x(t)$ "hidden" within it. Just as your eye has

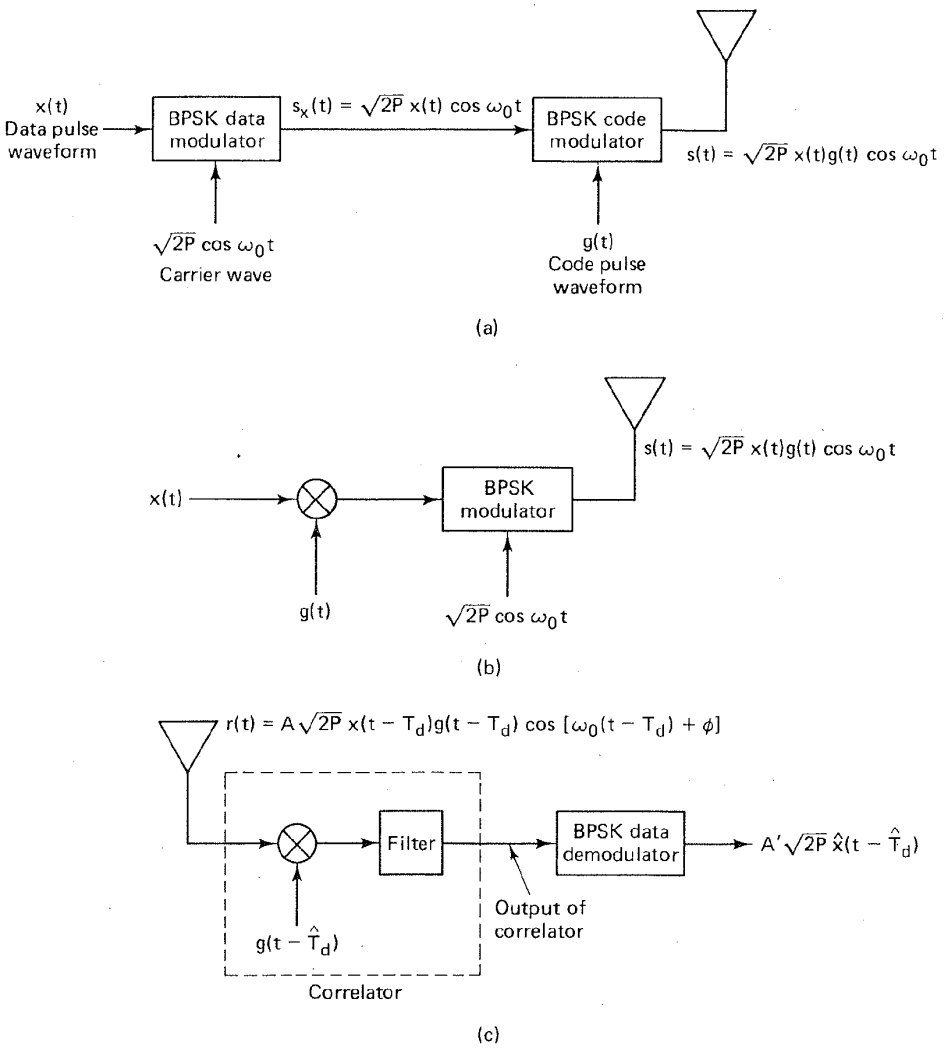


Figure 10.9 Direct-sequence spread-spectrum system. (a) BPSK direct-sequence transmitter. (b) Simplified BPSK direct-sequence transmitter. (c) BPSK direct-sequence receiver.

difficulty finding the slowly moving data signal in the rapidly moving code signal, it is similarly difficult for a receiver to recover a slowly moving signal from a rapidly moving code without having an exact replica of the code.

As shown in Figure 10.9c, DS/BPSK demodulation is a two-step process. The first step, despreading, is accomplished by correlating the received signal with a synchronized replica of the code. The second step, data demodulation, is accomplished with a conventional demodulator. In the example of Figure 10.10

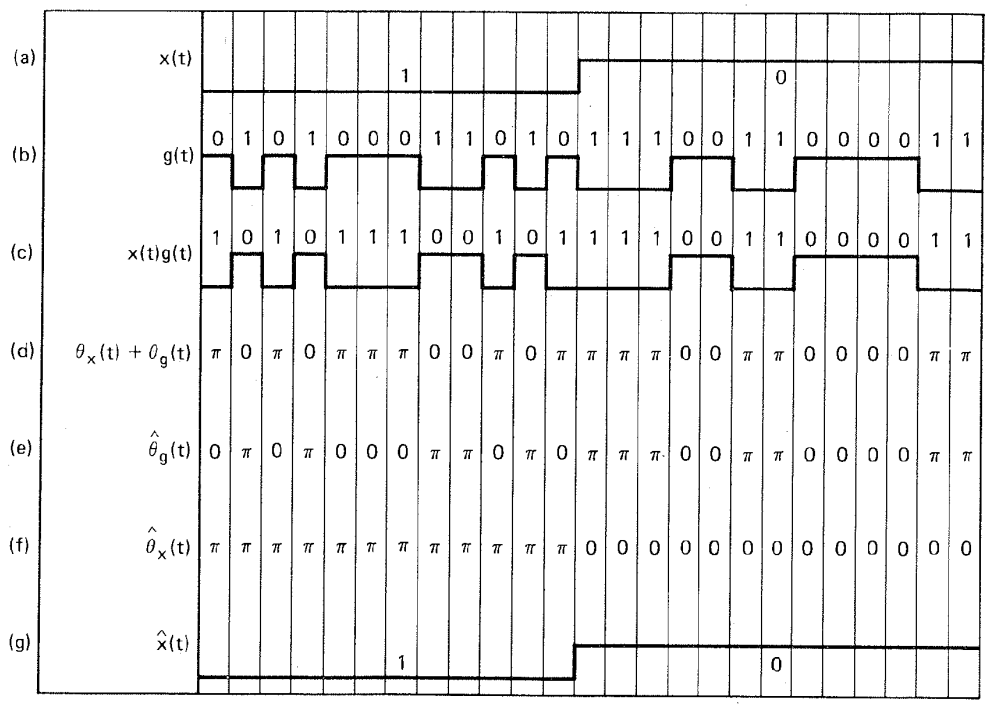


Figure 10.10 Spread-spectrum example using direct sequencing. (a) Binary data waveform to be transmitted. (b) Code sequence. (c) Transmitted sequence. (d) Phase of transmitted carrier. (e) Phase shift produced by receiver code. (f) Phase of received carrier after phase shift by receiver code. (g) Demodulated data waveform.

we see the code replica $\hat{\theta}_g(t)$, in Figure 10.10e, as the phase shift (either 0 or π) that is produced at the receiver by the despreading code. Figure 10.10f illustrates the resulting estimate of the carrier phase, $\hat{\theta}_x(t)$, after despreading or after $\hat{\theta}_g(t)$ has been added to $\theta_x(t) + \theta_g(t)$. At this point one can recognize the original data pattern in the phase terms of the carrier wave. The final step, shown in Figure 10.10g, is to recover an estimate of the data waveform, $\hat{x}(t)$, by the use of a BPSK demodulator.

10.3.2 Processing Gain and Performance

A fundamental issue in spread-spectrum systems is *how much* protection spreading can provide against interfering signals with finite power. Spread-spectrum techniques distribute a relatively low-dimensional signal in a large-dimensional signal space. The signal is "hidden" within the signal space, since we assume that a jammer does not know which signal coordinates are being transmitted at any time. The only recourse for the jammer, intent upon communication disruption, is to jam the entire space with its fixed total power, thus inducing a limited amount of interference in each signal coordinate, or to jam a portion of the signal

space with its total power, thus leaving the remainder of the signal space free of interference.

Consider a set of D orthogonal signals, $s_i(t)$, $1 \le i \le D$, in an N-dimensional space, where in general, $D \ll N$. Following the development in Section 3.2.2, we can write

$$s_i(t) = \sum_{j=1}^{N} a_{ij}\psi_j(t) \qquad \begin{matrix} i = 1, 2, \ldots, D; \quad 0 \le t \le T \\ D \ll N \end{matrix} \tag{10.12}$$

where

$$a_{ij} = \int_0^T s_i(t)\psi_j(t)\, dt \tag{10.13}$$

and

$$\int_0^T \psi_j(t)\psi_k(t)\, dt = \begin{cases} 1 & \text{for } j = k \\ 0 & \text{otherwise} \end{cases} \tag{10.14}$$

The $\{\psi_j(t)\}$ are linearly independent functions that *span* or characterize the N-dimensional orthonormal space and are called *basis* functions of the space. For every information symbol that is transmitted, a set of coefficients $\{a_{ij}\}$ is chosen independently, using a pseudorandom spreading code, in order to hide the D-dimensional signal set in the larger N-dimensional space. The set of random variables $\{a_{ij}\}$ assume the values $\pm a$, each with a probability of $\frac{1}{2}$. The receiver, of course, has access to each set of coefficients chosen in order to perform the necessary correlation despreading. Even if the same ith symbol is sent repeatedly, the set $\{a_{ij}\}$ used to transmit it is newly selected from symbol to symbol. The energy in each signal waveform of the D signal set will be assumed equal, so that we can write the average energy for each signal as follows:

$$E_s = \int_0^T \overline{s_i^2(t)}\, dt = \sum_{j=1}^{N} \overline{a_{ij}^2} \qquad i = 1, 2, \ldots, D \tag{10.15}$$

where the overbar means the expected value over the ensemble of many symbol transmissions. The independent coefficients have zero mean and correlation:

$$\overline{a_{ij}a_{ik}} = \begin{cases} \dfrac{E_s}{N} & \text{for } j = k \\ 0 & \text{otherwise} \end{cases} \tag{10.16}$$

The standard assumption is that the jammer has no a priori knowledge regarding the selection of the signaling coefficients $\{a_{ij}\}$. As far as the jammer is concerned, the coefficients are uniformly distributed over the N basis coordinates. If the jammer chooses to distribute its power uniformly over the total signal space, the jammer waveform $w(t)$ can be written

$$w(t) = \sum_{j=1}^{N} b_j\psi_j(t) \tag{10.17}$$

with total energy

$$E_w = \int_0^T w^2(t)\, dt = \sum_{j=1}^{N} b_j^2 \tag{10.18}$$

A reasonable goal for a jammer would be to devise a strategy for selecting the portions b_j^2, of its fixed total energy E_w so as to minimize the desired signal-to-noise ratio (SNR) at the receiver after demodulation.

At the receiver, the detector output (ignoring receiver noise),

$$r(t) = s_i(t) + w(t) \tag{10.19}$$

is correlated with the set of possible transmitted signals, so that the output of the ith correlator z_i is

$$z_i = \int_0^T r(t)s_i(t)\, dt = \sum_{j=1}^{N} \left(a_{ij}^2 + b_j a_{ij}\right) \tag{10.20}$$

The second term on the right side of Equation (10.20) averages to zero over the ensemble of all possible pseudorandom code sequences, since the set of random variables $\{a_{ij}\}$ assume the values $\pm a$, each with probability $\frac{1}{2}$. Therefore, given that $s_m(t)$ was transmitted, the expected value of the output of the ith correlator, $\mathbf{E}(z_i|s_m)$, can be written, following the development in References [7, 8],

$$\mathbf{E}(z_i|s_m) = \sum_{j=1}^{N} \overline{a_{ij}^2} = \begin{cases} E_s & \text{for } i = m \\ 0 & \text{otherwise} \end{cases} \tag{10.21}$$

In Equation (10.21), the term $\mathbf{E}(z_i|s_m)$ for $i = m$ is to be interpreted as follows. Given that $s_i(t)$ is to be transmitted, N coefficients a_{ij} $(1 \leq j \leq N)$ are chosen pseudorandomly (the receiver is assumed to have access to each choice of the a_{ij} for correlation despreading). Hence, in computing $\mathbf{E}(z_i|s_i)$, even though the ith information symbol is specified at the transmitter, the pattern of coefficients used to send it appears random (to the unauthorized receiver) for each transmission. Equation (10.21) presumes that the jammer has not been successful in its attempt to employ some clever tactics (described in Section 10.7).

Let us assume that all D signals are equally likely. Then the expected value at the output of any of the D correlators is

$$\mathbf{E}(z_i) = \frac{E_s}{D} \tag{10.22}$$

Similarly, using Equations (10.15) to (10.21), we compute var $(z_i|s_i)$, the variance at the output of the ith correlator, given that the ith signal was transmitted.

$$\begin{aligned} \text{var}\,(z_i|s_i) &= \sum_{j,k} b_j b_k \overline{a_{ij} a_{ik}} \\ &= \sum_{j=1}^{N} b_j^2 \overline{a_{ij}^2} \end{aligned} \tag{10.23}$$

$$= \sum_{j=1}^{N} b_j^2 \frac{E_s}{N}$$

$$= \frac{E_w E_s}{N} \tag{10.24}$$

For completeness, the variance at the output of the ith correlator, var $(z_i|s_m)$, given that the mth signal was transmitted, where $i \neq m$, can similarly be computed to be

$$\text{var}\,(z_i|s_m) = \frac{E_w E_s}{N} + \frac{E_s^2}{N} \tag{10.25}$$

The signal-to-jammer ratio (SJR) at the output of the ith correlator can be defined as

$$\text{SJR} = \sum_{m=1}^{D} \frac{\mathbf{E}^2(z_i|s_m)}{\text{var}\,(z_i|s_m)} P(s_m) = \frac{E_s^2/D}{E_w E_s/N} = \frac{E_s N}{E_w D} \tag{10.26}$$

where the probability of the mth signal $P(s_m) = 1/D$, since the signals are assumed to occur with equal probability, and where the signal energy and the jammer energy in the ith correlator are denoted by $\mathbf{E}^2(z_i)$ and var (z_i), respectively. Because of Equation (10.21), the only terms in the summation of Equation (10.26) not equal to zero are those for which $i = m$. The result is independent of the way in which the jammer chooses to distribute its energy. Therefore, regardless of how b_j is chosen, subject to $\sum_j b_j^2 = E_w$, the SJR in Equation (10.26) indicates that spreading gives the signal an advantage of a factor of N/D over the jammer. The ratio N/D is known as the *processing gain* G_p.

Since the approximate dimensionality of a signal with bandwidth W and duration T is $2WT$, we can express the processing gain as

$$G_p = \frac{N}{D} \simeq \frac{2W_{ss}T}{2W_{\min}T} = \frac{W_{ss}}{R} \tag{10.27}$$

where W_{ss} is the spread-spectrum bandwidth (the total bandwidth used by the spreading technique) and $W_{\min}$ is the minimum bandwidth of the data (taken to be the data rate, R). For direct sequence systems, W_{ss} is approximately the code chip rate R_p, and $W_{\min}$ is similarly the data rate R, giving

$$G_p = \frac{R_p}{R} \tag{10.28}$$

10.4 FREQUENCY HOPPING SYSTEMS

We now consider a spread-spectrum technique called frequency hopping (FH). The modulation most commonly used with this technique is M-ary frequency shift keying (MFSK), where $k = \log_2 M$ information bits are used to determine which

one of M frequencies is to be transmitted. The position of the M-ary signal set is shifted pseudorandomly by the frequency synthesizer over a hopping bandwidth W_{ss}. A typical FH/MFSK system block diagram is shown in Figure 10.11. In a conventional MFSK system, the data symbol modulates a *fixed frequency* carrier; in an FH/MFSK system, the data symbol modulates a carrier whose frequency is *pseudorandomly* determined. In either case, a single tone is transmitted. The FH system in Figure 10.11 can be thought of as a two-step modulation process—data modulation and frequency hopping modulation—even though it can be implemented as a single step whereby the frequency synthesizer produces a transmission tone based on the simultaneous dictates of the PN code and the data. At each frequency hop time a PN generator feeds the frequency synthesizer a frequency word (a sequence of ℓ chips) which dictates one of 2^{ℓ} symbol-set positions. The frequency hopping bandwidth, W_{ss}, and the minimum frequency spacing between consecutive hop positions, Δf, dictate the minimum number of chips necessary in the frequency word.

For a given hop, the occupied transmission bandwidth is identical to the bandwidth of conventional MFSK, which is typically much smaller than W_{ss}. However, averaged over many hops, the FH/MFSK spectrum occupies the entire spread-spectrum bandwidth. Current technology permits FH bandwidths of the order of several gigahertz, which is an order of magnitude larger than implementable DS bandwidths [9], thus allowing for larger processing gains in FH compared to DS systems. Since frequency hopping techniques operate over such wide bandwidths, it is difficult to maintain phase coherence from hop to hop. Therefore, such schemes are usually configured using noncoherent demodulation. Nevertheless, consideration has been given to coherent FH in Reference [10].

In Figure 10.11 we see that the receiver reverses the signal processing steps of the transmitter. The received signal is first FH demodulated (dehopped) by mixing it with the same sequence of pseudorandomly selected frequency tones that was used for hopping. Then the dehopped signal is applied to a conventional bank of M noncoherent energy detectors to select the most likely symbol.

Example 10.1 Frequency Word Size

A hopping bandwidth W_{ss} of 400 MHz and a frequency step size Δf of 100 Hz are specified. What is the minimum number of PN chips that are required for each frequency word?

Solution

$$\text{Number of tones contained in } W_{ss} = \frac{W_{ss}}{\Delta f} = \frac{400 \text{ MHz}}{100 \text{ Hz}}$$

$$= 4 \times 10^6$$

$$\text{Minimum number of chips} = \lceil \log_2 (4 \times 10^6) \rceil$$

$$= 22 \text{ chips}$$

where $\lceil x \rceil$ indicates the smallest integer value not less than x.

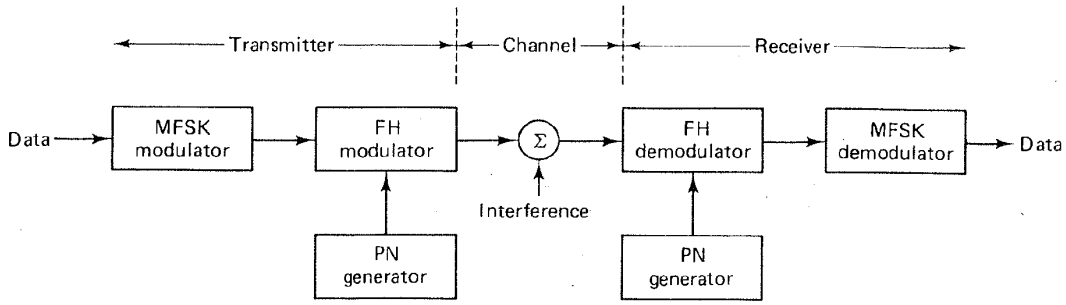


Figure 10.11 FH/MFSK system.

10.4.1 Frequency Hopping Example

Consider the frequency hopping example illustrated in Figure 10.12. The input data consist of a binary sequence with a data rate of $R = 150$ bits/s. The modulation is 8-ary FSK. Therefore, the symbol rate is $R_s = R/(\log_2 8) = 50$ symbols/s (the symbol duration $T = 1/50 = 20$ ms). The frequency is hopped once per symbol, and the hopping is time-synchronous with the symbol boundaries. Thus the hopping rate is 50 hops/s. Figure 10.12 depicts the time–bandwidth plane of the communication resource; the abscissa represents time, and the ordinate represents

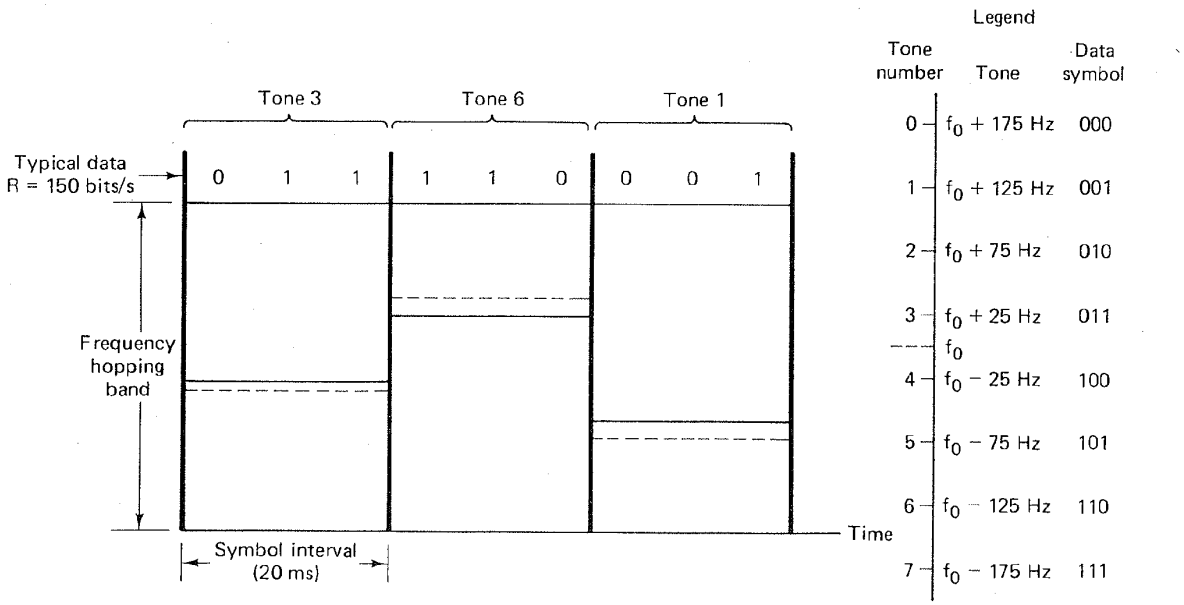


Figure 10.12 Frequency hopping example using 8-ary FSK modulation.

the hopping bandwidth, W_{ss}. The legend on the right side of the figure illustrates a set of 8-ary FSK symbol-to-tone assignments. Notice that the tone separation specified is $1/T = 50$ Hz, which corresponds to the minimum required tone spacing for the orthogonal signaling of this noncoherent FSK example (see Section 3.6.4).

A typical binary data sequence is shown at the top of Figure 10.12. Since the modulation is 8-ary FSK, the bits are grouped three at a time to form symbols. In a *conventional* 8-ary FSK scheme, a single-sideband tone, offset from f_0, the *fixed* center frequency of the data band, would be transmitted (according to an assignment like the one shown in the legend). The only difference in this FH/MFSK example is that the center frequency of the data band, f_0, is *not fixed*. For each new symbol, f_0 hops to a new position in the hop bandwidth, and the entire data-band structure moves with it. In the example of Figure 10.12, the first symbol in the data sequence, 0 1 1, yields a tone 25 Hz above f_0. The diagram depicts f_0 with a dashed line and the symbol tone with a solid line. During the second symbol interval, f_0 has hopped to a new spectral location, as indicated by the dashed line. The second symbol, 1 1 0, dictates that a tone indicated by the solid line, 125 Hz below f_0, shall be transmitted. Similarly, the final symbol in this example, 0 0 1, calls for a tone 125 Hz above f_0. Again, the center frequency has moved, but the relative positions of the symbol tones remain fixed.

10.4.2 Robustness

A common dictionary definition describes the term *robustness* as the state of being strong and healthy; full of vigor; hardy. In the context of communications, the usage is not too different. Robustness characterizes a signal's ability to withstand impairments from the channel, such as noise, jamming, fading, and so on. A signal configured with multiple replicate copies, each transmitted on a different frequency, has a greater likelihood of survival than does a single such signal with equal total power. The greater the diversity (multiple transmissions, at different frequencies, spread in time), the more robust the signal against random interference.

The following example should clarify the concept. Consider a message consisting of four symbols s_1, s_2, s_3, s_4. The introduction of diversity starts by repeating the message N times. Let us choose $N = 8$. Then, the repeated symbols called *chips* can be written

$$s_1s_1s_1s_1s_1s_1s_1s_1s_2s_2s_2s_2s_2s_2s_2s_2s_3s_3s_3s_3s_3s_3s_3s_3s_4s_4s_4s_4s_4s_4s_4s_4$$

Each chip is transmitted at a different hopping frequency (the center of the data bandwidth is changed for each chip). The resulting transmissions at frequencies f_i, f_j, f_k, . . . yield a more robust signal than without such divesity. A target-shooting analogy is that a pellet from a barrage of shotgun pellets has a better chance of hitting a target, compared to the action of a single bullet.

10.4.3 Frequency Hopping with Diversity

In Figure 10.13 we extend the example illustrated in Figure 10.12, with the additional feature of a chip repeat factor of $N = 4$. During each 20-ms symbol interval, there are now four columns, corresponding to the four separate chips to be transmitted for each symbol. At the top of the figure we see the same data sequence, with $R = 150$ bps, as in the earlier example; and we see the same 3-bit partitioning to form the 8-ary symbols. Each symbol is transmitted four times, and for each transmission the center frequency of the data band is hopped to a new region of the hopping band, under the control of a PN code generator. Therefore, for this example, each chip interval, T_c, is equal to $T/N = 20$ ms/4 = 5 ms in duration, and the hopping rate is now

$$\frac{NR}{\log_2 8} = 200 \text{ hops/s}$$

Notice that the spacing between frequency tones must change to meet the changed requirement for orthogonality. Since the duration of each FSK tone is now equal to the chip duration, that is, $T_c = T/N$, the minimum separation between tones is $1/T_c = N/T = 200$ Hz. As in the earlier example, Figure 10.13 illustrates that the center of the data band (plus the modulation structure) is shifted at each new

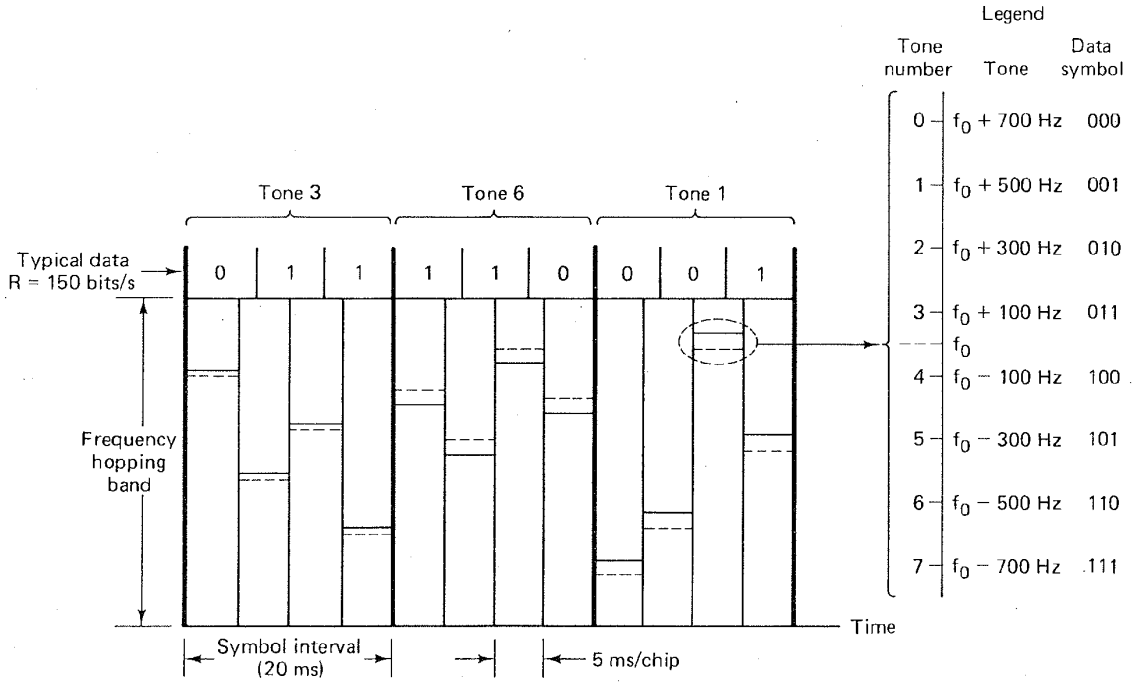


Figure 10.13 Frequency hopping example with diversity ($N = 4$).

chip time. The position of the solid line (transmission frequency) has the same relationship to the dashed line (center of the data band) for each of the chips associated with a given symbol.

10.4.4 Fast Hopping versus Slow Hopping

In the case of direct-sequence spread-spectrum systems, the term "chip" refers to the PN code symbol (the symbol of shortest duration in a DS system). In a similar sense for frequency hopping systems, the term "chip" is used to characterize the shortest uninterrupted waveform in the system. Frequency hopping systems are classified as *slow frequency hopping* (SFH), which means there are several modulation symbols per hop, or as *fast frequency hopping* (FFH), which means that there are several frequency hops per modulation symbol. For SFH, the shortest uninterrupted waveform in the system is that of the data symbol; however, for FFH, the shortest uninterrupted waveform is that of the hop. Figure 10.14a illustrates an example of FFH; the data symbol rate is 30 symbols/s and the frequency hopping rate is 60 hops/s. The figure illustrates the waveform $s(t)$ over one symbol duration ($\frac{1}{30}$ s). The waveform change in (the middle of) $s(t)$ is

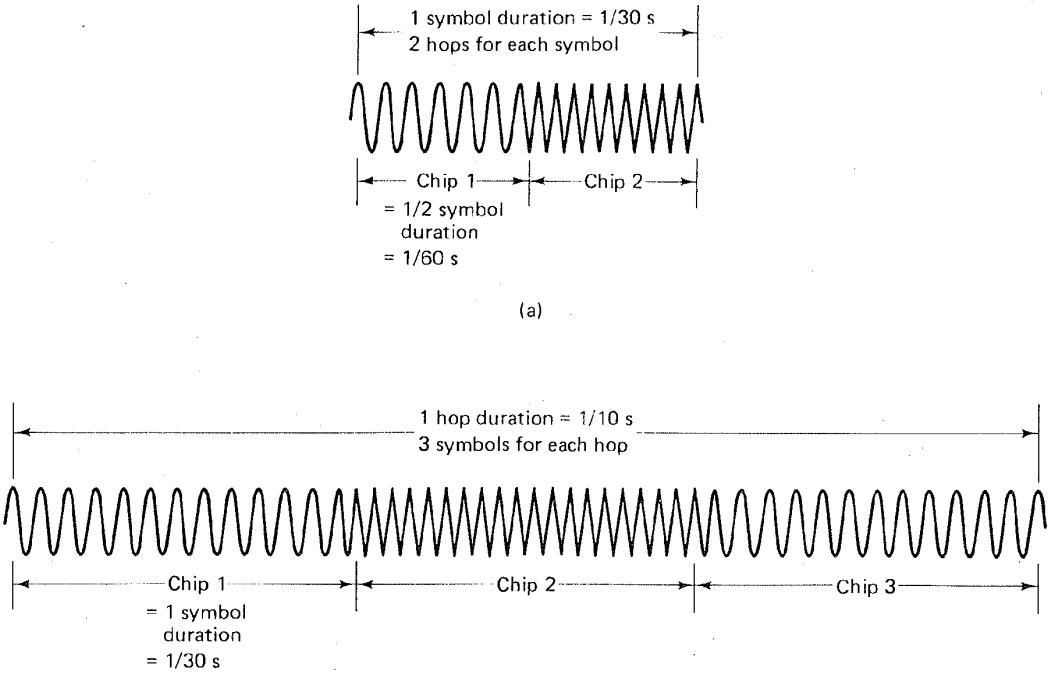


Figure 10.14 Chip—in the context of an FH/MFSK system. (a) Example 1: Frequency hopping MFSK system with symbol rate = 30 symbols/s and hopping rate = 60 hops/s. 1 chip = 1 hop. (b) Example 2: Same as part (a) except hopping rate = 10 hops/s. 1 chip = 1 symbol.

due to a new frequency hop. In this example, a chip corresponds to a hop since the hop duration is shorter than the symbol duration. Each chip corresponds to half a symbol. Figure 10.14b illustrates an example of SFH; the data symbol rate is still 30 symbols/s, but the frequency hopping rate has been reduced to 10 hops/s. The waveform $s(t)$ is shown over a duration of three symbols ($\frac{1}{10}$ s). In this example, the hopping boundaries appear only at the beginning and end of the three-symbol duration. Here, the changes in the waveform are due to the modulation state changes; therefore, in this example a chip corresponds to a data symbol, since the data symbol is shorter than the hop duration.

Figure 10.15a illustrates an FFH example of a binary FSK system. The diversity is $N = 4$. There are 4 chips transmitted per bit. As in Figure 10.13, the dashed line in each column corresponds to the center of the data band and the solid line corresponds to the symbol frequency. Here, for FFH, the chip duration is the hop duration. Figure 10.15b illustrates an example of an SFH binary FSK system. In this case, there are 3 bits transmitted during the time duration of a single hop. Here, for SFH, the chip duration is the bit duration. If this SFH example were changed from a binary system to an 8-ary system, what would the

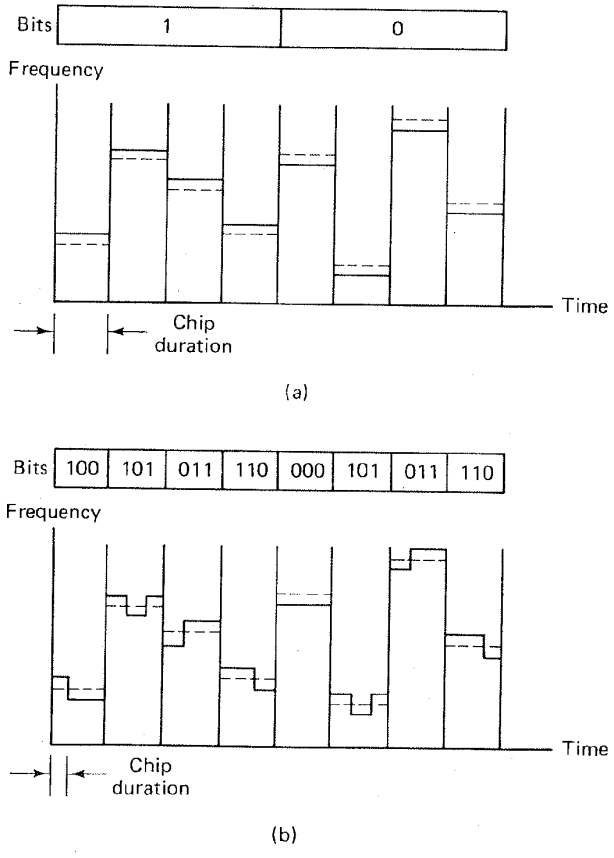


Figure 10.15 Fast hopping versus slow hopping in a binary system. (a) Fast-hopping example: 4 hops/bit. (b) Slow-hopping example: 3 bits/hop.

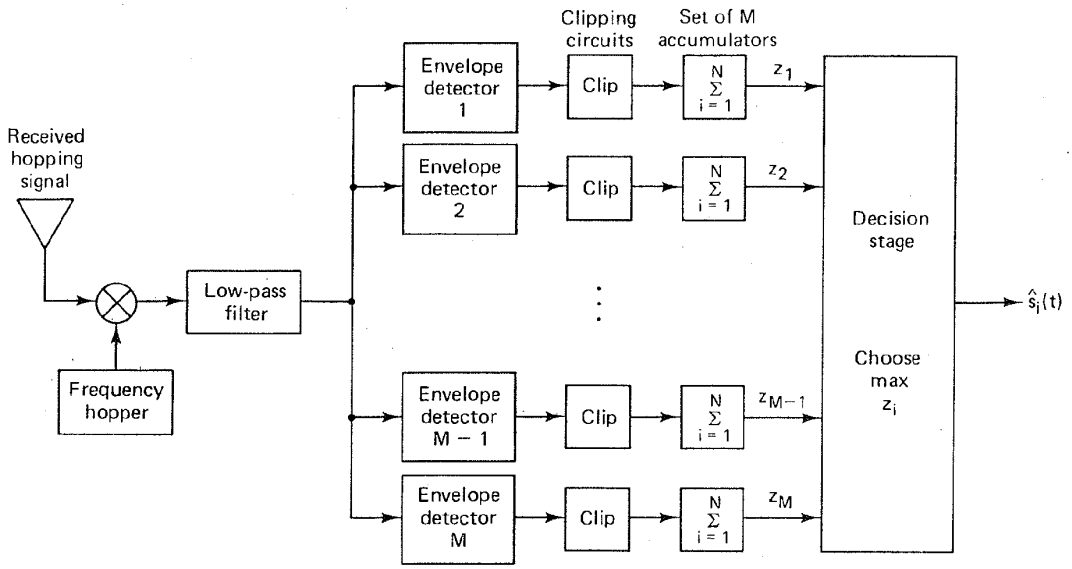


Figure 10.16 FFH/MFSK demodulator.

chip duration then correspond to? If the system were implemented as an 8-ary scheme, each 3 bits would be transmitted as a single data symbol. The symbol boundaries and the hop boundaries would then be the same, and the chip duration, the hop duration, and the symbol duration would all be the same.

10.4.5 FFH/MFSK Demodulator

Figure 10.16 illustrates the schematic for a typical fast frequency hopping MFSK (FFH/MFSK) demodulator. First, the signal is dehopped using a PN generator identical to the one used for hopping. Then, after filtering with a low-pass filter that has a bandwidth equal to the data bandwidth, the signal is demodulated using a bank of M envelope or energy detectors. Each envelope detector is followed by a clipping circuit and an accumulator. The clipping circuit serves an important function in the presence of an intentional jammer or other strong unpredictable interference; it is treated in a later section. The demodulator does *not* make symbol decisions on a chip-by-chip basis. Instead, the energy from the N chips are accumulated, and after the energy from the Nth chip is added to the $N - 1$ earlier ones, the demodulator makes a symbol decision by choosing the symbol that corresponds to the accumulator, z_i ($i = 1, 2, \ldots, M$), with maximum energy.

10.5 SYNCHRONIZATION

For both DS and FH spread-spectrum systems, a receiver must employ a *synchronized* replica of the spreading or code signal to demodulate the received signal successfully. The process of synchronizing the locally generated spreading signal with the received spread-spectrum signal is usually accomplished in two steps.

The first step, called *acquisition,* consists of bringing the two spreading signals into *coarse* alignment with one another. Once the received spread-spectrum signal has been acquired, the second step, called *tracking,* takes over and continuously maintains the best possible waveform *fine* alignment by means of a feedback loop.

10.5.1 Acquisition

The acquisition problem is one of searching throughout a region of time and frequency uncertainty in order to synchronize the received spread-spectrum signal with the locally generated spreading signal. Acquisition schemes can be classified as coherent or noncoherent. Since the despreading process typically takes place before carrier synchronization, and therefore the carrier phase is unknown at this point, most acquisition schemes utilize noncoherent detection. When determining the limits of the uncertainty in time and frequency, the following items must be considered:

1. Uncertainty in the distance between the transmitter and the receiver translates into uncertainty in the amount of propagation delay.
2. Relative clock instabilities between the transmitter and the receiver result in phase differences between the transmitter and receiver spreading signals that will tend to grow as a function of elapsed time between synchronization.
3. Uncertainty of the receiver's relative velocity with respect to the transmitter translates into uncertainty in the value of Doppler frequency offset of the incoming signal.
4. Relative oscillator instabilities between the transmitter and the receiver result in frequency offsets between the two signals.

10.5.1.1 Correlator Structures

A common feature of all acquisition methods is that the received signal and the locally generated signal are first correlated to produce a measure of similarity between the two. This measure is then compared to a threshold to decide if the two signals are in synchronism. If they are, the tracking loop takes over.* If they are not, the acquisition procedure provides for a phase or frequency change in the locally generated code as a part of a systematic search through the receiver's phase and frequency uncertainty region, and another correlation is attempted.

Consider the direct-sequence *parallel-search* acquisition system shown in Figure 10.17. The locally generated code $g(t)$ is available with delays that are spaced one-half chip ($T_c/2$) apart. If the time uncertainty between the local code and the received code is N_c chips and a complete parallel search of the entire time uncertainty region is to be accomplished in a single search time, $2N_c$ correlators are used. Each correlator simultaneously examines a sequence of λ chips, after which the $2N_c$ correlator outputs are compared. The locally generated code, corresponding to the correlator with the largest output is chosen. Conceptually, this is the simplest of the search techniques; it considers all possible code positions

* Quite often to maintain a small false alarm probability, the threshold crossing must be further verified by a suitable verification algorithm before the tracking loop takes over [4].

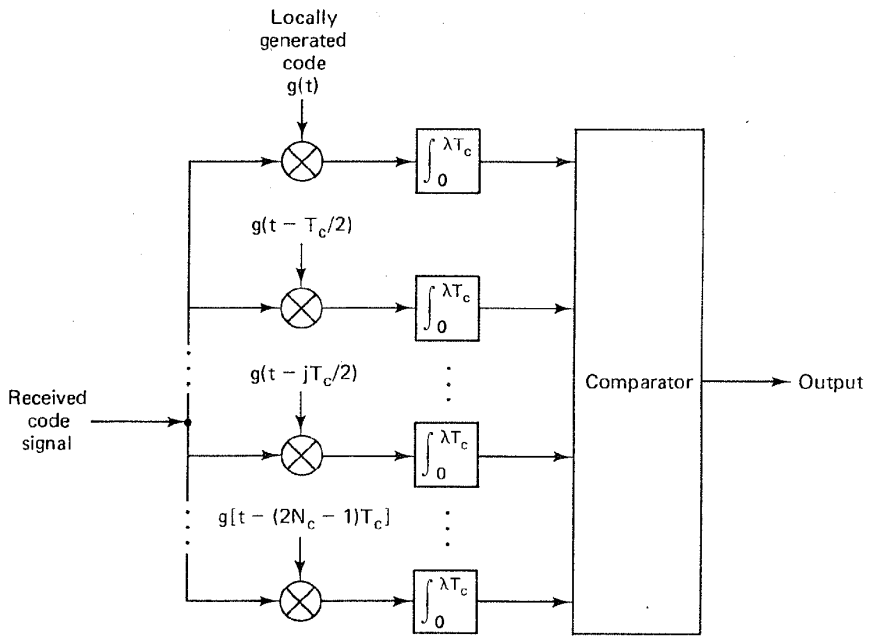


Figure 10.17 Direct-sequence parallel search acquisition.

(or fractional code positions) in parallel and uses a maximum likelihood algorithm for acquiring the code. Each detector output pertains to the identical observation of received signal plus noise. As λ increases, the synchronization error probability (i.e., the probability of choosing the incorrect code alignment) decreases. Thus λ is chosen as a compromise between minimizing the probability of a synchronization error and minimizing the time to acquire.

Figure 10.18 illustrates a simple acquisition scheme for a frequency hopping system. Assume that a sequence of N consecutive frequencies from the hop sequence is chosen as a synchronization pattern (without data modulation). The N noncoherent matched filters each consists of a mixer followed by a bandpass filter (BPF) and a square-law envelope detector (an envelope detector followed by a square-law device). If the frequency hopping sequence is $f_1, f_2, \ldots, f_N$, delays are inserted into the matched filters so that when the correct frequency hopping sequence appears, the system produces a large output, indicating detection of the synchronization sequence. Acquisition can be accomplished rapidly because all possible code offsets are examined simultaneously.

If, during each correlation, λ chips are examined, the maximum time required, $(T_{\text{acq}})_{\max}$, for a fully parallel search is

$$(T_{\text{acq}})_{\max} = \lambda T_c \tag{10.29}$$

The mean acquisition time of a parallel search system can be approximated by noting that after integrating over λ chips, a correct decision will be made with probability P_D, called the *probability of detection*. If an incorrect output is chosen,

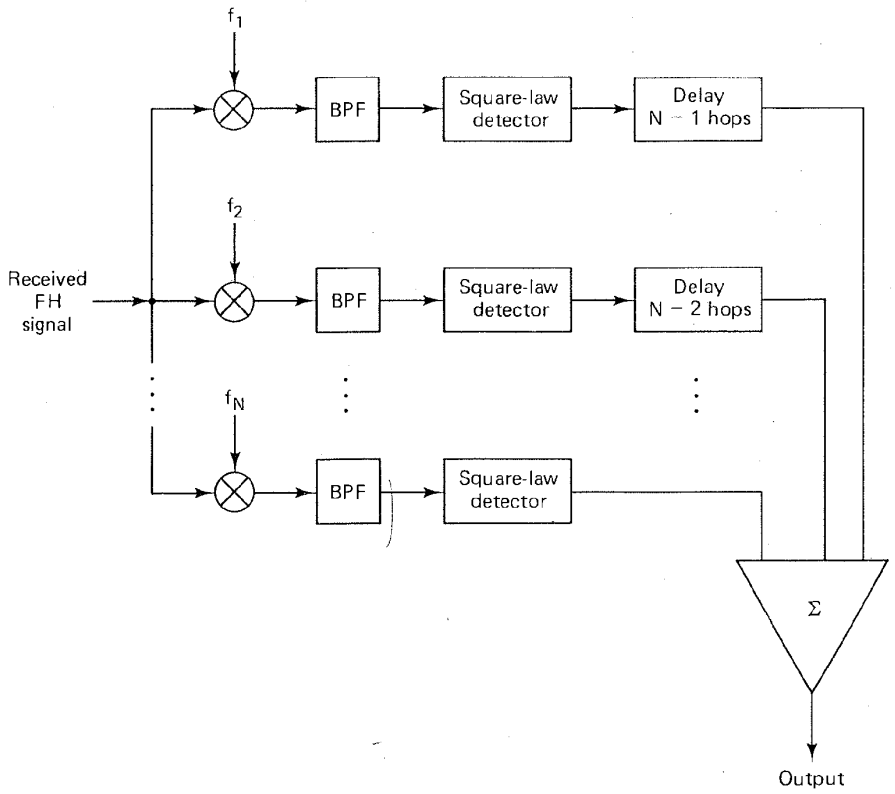


Figure 10.18 Frequency hopping acquisition scheme.

an additional λ chips are again examined to make a determination of the correct output. Therefore, on the average, the acquisition time is [6]

$$\overline{T}_{\text{acq}} = \lambda T_c P_D + 2\lambda T_c P_D(1 - P_D) + 3\lambda T_c P_D(1 - P_D)^2 + \cdots$$

$$= \frac{\lambda T_c}{P_D} \tag{10.30}$$

Since the required number of correlators or matched filters can be prohibitively large, fully parallel acquisition techniques are not usually used. In place of Figures 10.17 and 10.18, a single correlator or matched filter can be implemented that will *serially search* until synchronization is achieved. Naturally, trade-offs between fully parallel, fully serial, and combinations of the two involve hardware complexity versus time to acquire for the same uncertainty and chip rate.

10.5.1.2 Serial Search

A popular strategy for the acquisition of spread-spectrum signals is to use a single correlator or matched filter to serially search for the correct phase of the DS code signal or the correct hopping pattern of the FH signal. A considerable

reduction in complexity, size, and cost can be achieved by a serial implementation that repeats the correlation procedure for each possible sequence shift. Figures 10.19 and 10.20 illustrate the basic configuration for DS and FH spread-spectrum schemes, respectively. In a stepped serial acquisition scheme for a DS system, the timing epoch of the local PN code is set, and the locally generated PN signal is correlated with the incoming PN signal. At fixed examination intervals of λT_c (search dwell time), where $\lambda \gg 1$, the output signal is compared to a preset threshold. If the output is below the threshold, the phase of the locally generated code signal is incremented by a fraction (usually one-half) of a chip and the correlation is reexamined. When the threshold is exceeded, the PN code is assumed to have been acquired, the phase-incrementing process of the local code is inhibited, and the code tracking procedure will be initiated. In a similar scheme for FH systems, shown in Figure 10.20, the PN code generator controls the frequency hopper. Acquisition is accomplished when the local hopping is aligned with that of the received signal.

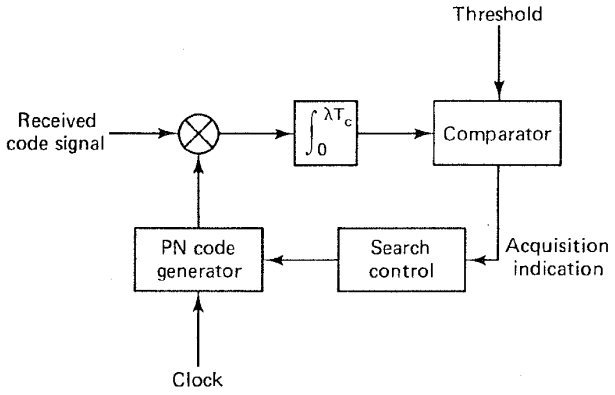


Figure 10.19 Direct-sequence serial search acquisition.

The maximum time required for a fully serial DS search, assuming that the search proceeds in half-chip increments, is

$$(T_{\text{acq}})_{\max} = 2N_c\lambda T_c \tag{10.31}$$

where the uncertainty region to be searched is N_c chips long. The mean acquisition time of a serial DS search system can be shown, for $N_c \gg \frac{1}{2}$ chip, to be [4]

$$\bar{T}_{\text{acq}} = \frac{(2 - P_D)(1 + KP_{\text{FA}})}{P_D}(N_c\lambda T_c) \tag{10.32}$$

where λT_c is the search dwell time, P_D the probability of correct detection, and P_{FA} the probability of false alarm. We can regard the time interval $K\lambda T_c$, where $K \gg 1$, as the time needed to verify a detection. Therefore, in the event of a false alarm, $K\lambda T_c$ seconds is the time penalty incurred. For $N_c \gg \frac{1}{2}$ chip and $K \ll 2N_c$, the variance of the acquisition time is

$$(\text{var})_{\text{acq}} = (2N_c\lambda T_c)^2(1 + KP_{\text{FA}})\left(\frac{1}{12} + \frac{1}{P_D^2} - \frac{1}{P_D}\right) \tag{10.33}$$

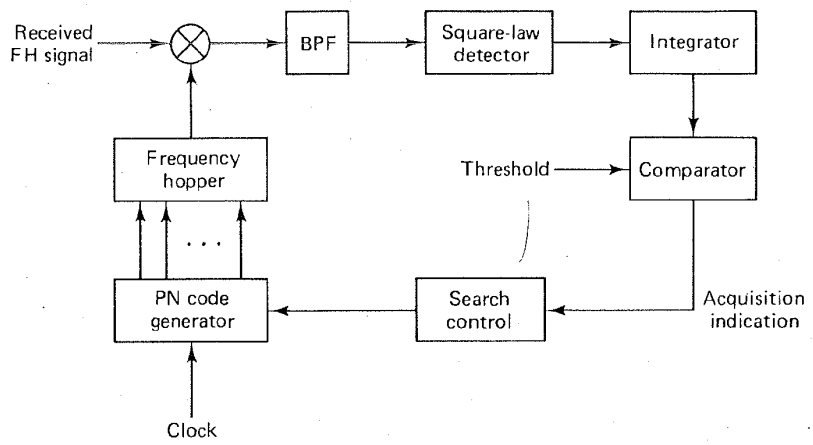


Figure 10.20 Frequency hopping serial search acquisition.

10.5.1.3 Sequential Estimation

Another search technique, called *rapid acquisition by sequential estimation* (RASE), proposed by Ward [11], is illustrated in Figure 10.21. The switch is initially in position 1. The RASE system enters its best estimate of the first n received code chips into the n stages of its local PN generator. The fully loaded register defines a starting state from which the generator begins its operation. A PN sequence has the property that the next combination of register states depends only on the present combination of states. Therefore, if the first n received chips are correctly estimated, all the following chips from the local PN generator will

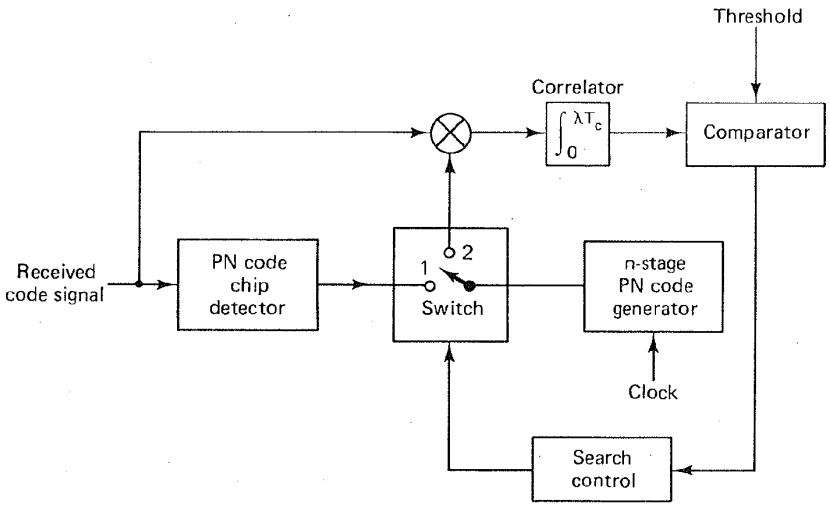


Figure 10.21 Rapid acquisition by sequential estimation.

be correctly generated. The switch is next thrown to position 2. If the starting state had been correctly estimated, the local generator generates the same sequence as the incoming waveform, in the absence of noise. If the correlator output after λT_c exceeds a preset threshold level, we assume that synchronization has occurred. If the output is less than the threshold, the switch is returned to position 1, the register is reloaded with estimates of the next n received chips, and the procedure is repeated. Once synchronization has occurred, the system no longer needs estimates of the input code chips. We can calculate the *minimum* acquisition time for the case when no noise is present. The first n chips will be correctly loaded into the register, and therefore the acquisition time is

$$T_{\text{acq}} = nT_c \tag{10.34}$$

While the RASE system has a rapid acquisition capability it has the drawback of being highly vulnerable to noise and interference signals. The reason for this is that the estimation process consists of a simple chip-by-chip hard-decision demodulation, without using the interference rejection benefits of the PN code.

For an extensive treatment of sequential estimation, see Reference [4].

10.5.2 Tracking

Once acquisition or coarse synchronization is completed, tracking or fine synchronization takes place. Tracking code loops can be classified as coherent or noncoherent. A coherent loop is one in which the carrier frequency and phase are known exactly so that the loop can operate on a baseband signal. A noncoherent loop is one in which the carrier frequency is not known exactly (due to Doppler effects, for example), nor is the phase. In most instances, since the carrier frequency and phase are not known exactly, a priori, a noncoherent code loop is used to track the received PN code. Tracking loops are further classified as a *full-time* early-late tracking loop, often referred to as a *delay-locked loop* (DLL), or as a *time-shared* early-late tracking loop, frequently referred to as a *tau-dither loop* (TDL). A basic noncoherent DLL loop for a direct-sequence spread-spectrum system using binary phase shift keying (BPSK) is shown in Figure 10.22. The data $x(t)$ and the code $g(t)$ each modulate the carrier wave using BPSK, and as before in the absence of noise and interference, the received waveform can be expressed as

$$r(t) = A\sqrt{2P}\, x(t)g(t) \cos(\omega_0 t + \phi) \tag{10.35}$$

where the constant A is a system gain parameter and ϕ is a random phase angle in the range $(0, 2\pi)$. The locally generated code of the tracking loop is offset in phase from the incoming $g(t)$ by a time τ, where $\tau < T_c/2$. The loop provides *fine* synchronization by first generating two PN sequences $g(t + T_c/2 + \tau)$ and $g(t - T_c/2 + \tau)$ delayed from each other by one chip. The two bandpass filters are designed to pass the data and to average the product of $g(t)$ and the two PN sequences $g(t \pm T_c/2 + \tau)$. (See Reference [4] for the optimum filter bandwidth for a given filter type.) The square-law envelope detector eliminates the data since $|x(t)| = 1$. The output of each envelope detector is given approximately by

$$E_D \simeq \mathbf{E}\left\{\left| g(t)g\left(t \pm \frac{T_c}{2} + \tau\right)\right|\right\} = \left|R_g\left(\tau \pm \frac{T_c}{2}\right)\right| \qquad (10.36)$$

where the operator $\mathbf{E}\{\cdot\}$ means *expected value* and $R_g(x)$ is the autocorrelation function of the PN waveform as shown in Figure 10.8. The feedback signal $Y(\tau)$ is shown in Figure 10.23. When τ is positive, the feedback signal $Y(\tau)$ instructs the voltage-controlled oscillator (VCO) to increase its frequency, thereby forcing τ to decrease, and when τ is negative, $Y(\tau)$ instructs the VCO to decrease, thereby forcing τ to increase. When τ is a suitably small number, $g(t)g(t + \tau) \simeq 1$, yielding the despread signal $Z(t)$, which is then applied to the input of a conventional data demodulator. Detailed analysis of the DLL can be found in References [4, 12–14].

A problem with the DLL is that the early and late arms must be precisely gain balanced or else the feedback signal $Y(\tau)$ will be offset and will not produce a zero signal when the error is zero. This problem is solved by using a time-shared tracking loop in place of the full-time delay-locked loop. The time-shared loop time shares the use of the early-late correlators. The main advantages are that only one correlator need be used in the design of the loop, and further, that dc offset problems are reduced.

An offshoot of the time-shared tracking loop is called the *tau-dither loop* (TDL), shown in Figure 10.24. This design has the advantage that only one correlator is needed to provide the code *tracking* function *and* the *despreading* function. Just as in the case of a DLL, the received signal is correlated with an early and a late version of the locally generated PN code. As shown in Figure 10.24, the PN code generator is driven by a clock signal whose phase is *dithered* back

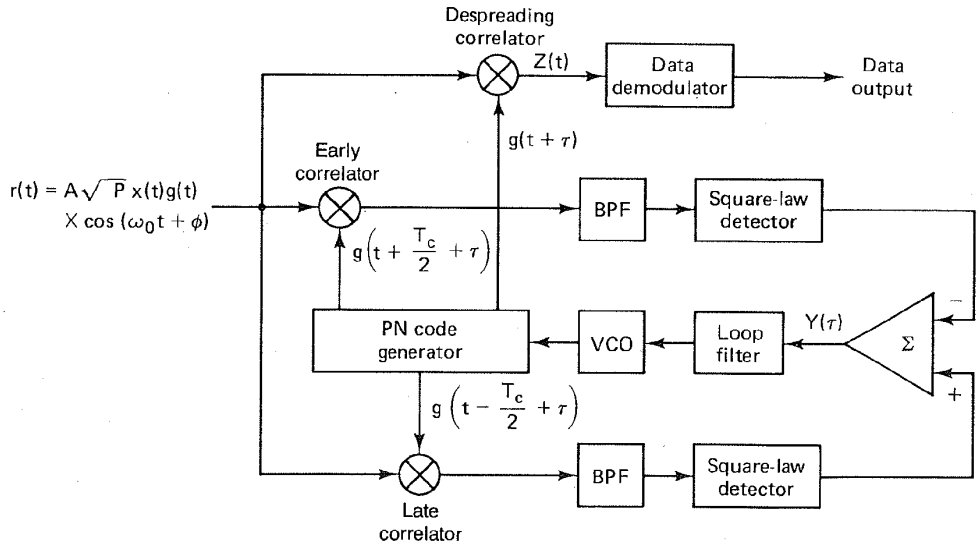


Figure 10.22 Delay-locked loop for tracking direct-sequence signals.

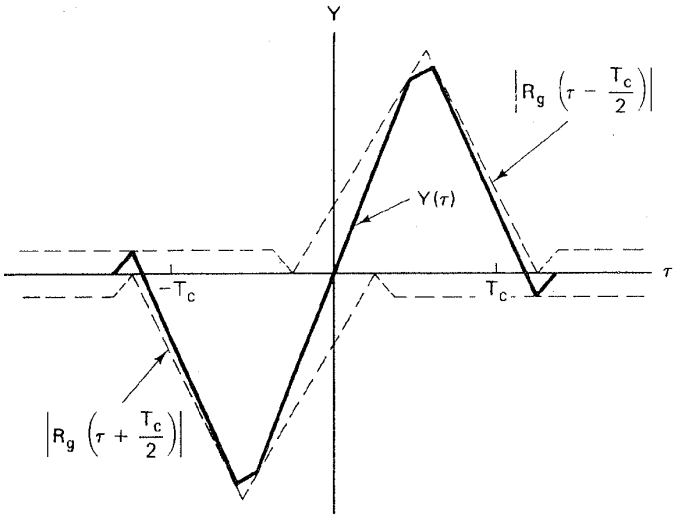


Figure 10.23 DLL feedback signal $Y(\tau)$.

and forth with a square-wave switching function; this eliminates the necessity of ensuring identical transfer functions of the early and late paths. The signal-to-noise performance of the TDL is only about 1.1 dB worse than that of the DLL if the arm filters are designed properly [4]. For a comprehensive treatment of synchronization of PN codes, see References [4, 15,16].

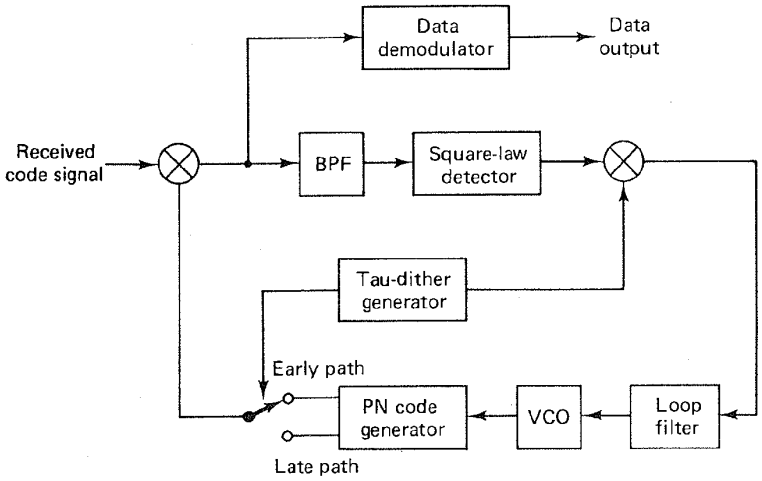


Figure 10.24 Tau-dither tracking loop.

10.6 SPREAD-SPECTRUM APPLICATIONS

10.6.1 Code-Division Multiple Access

Spread-spectrum multiple access techniques allow multiple signals occupying the same RF bandwidth to be transmitted simultaneously without interfering with one another. The application of spread-spectrum techniques to the problem of multiple access was discussed in Chapter 9 for a frequency hopped code-division multiple access (FH/CDMA) scheme. Here we consider CDMA using direct sequence (DS/CDMA). In these schemes, each of N user groups is given its own code, $g_i(t)$, where $i = 1, 2, \ldots, N$. The user codes are approximately orthogonal, so that the cross-correlation of two different codes is near zero. The main advantage of a CDMA system is that all the participants can share the full spectrum of the resource asynchronously; that is, the transition times of the different users' symbols do not have to coincide.

A typical DS/CDMA block diagram is shown in Figure 10.25. The first block illustrates the data modulation of a carrier, $A \cos \omega_0 t$. The output of the data modulator belonging to a user from group 1, $s_1(t)$, is shown below. The waveform is very general in form; no restriction has been placed on the type of modulation that can be used.

$$s_1(t) = A_1(t) \cos [\omega_0 t + \phi_1(t)] \tag{10.37}$$

Next, the data-modulated signal is multiplied by the spreading signal $g_1(t)$ belonging to user group 1, and the resulting signal, $g_1(t)s_1(t)$, is transmitted over the channel. Simultaneously, users from group 2 through N multiply their signals by their own code functions. Frequently, each code function is kept secret, and its use is restricted to the community of authorized users. The signal present at the receiver is the linear combination of the emanations from each of the users. Neglecting signal delays, we show this linear combination below.

$$g_1(t)s_1(t) + g_2(t)s_2(t) + \cdots + g_N(t)s_N(t) \tag{10.38}$$

As mentioned earlier, multiplication of $s_1(t)$ by $g_1(t)$ produces a signal whose spectrum is the convolution of the spectrum of $s_1(t)$ with the spectrum of $g_1(t)$. Thus, assuming that the signal $s_1(t)$ is relatively narrowband compared with the code or spreading signal $g_1(t)$, the product signal $g_1(t)s_1(t)$ will have approximately the bandwidth of $g_1(t)$. Assume that the receiver is configured to receive messages from user group 1. Assume, too, that the $g_1(t)$ code, generated at the receiver, is perfectly synchronized with the received signal from a group 1 user. The first stage of the receiver multiplies the incoming signal of Equation (10.38) by $g_1(t)$. The output of the multiplier will yield the following terms:

Desired signal: $g_1^2(t)s_1(t)$

Plus a composite of

undesired signals: $g_1(t)g_2(t)s_2(t) + g_1(t)g_3(t)s_3(t) + \cdots + g_1(t)g_N(t)s_N(t)$ (10.39)

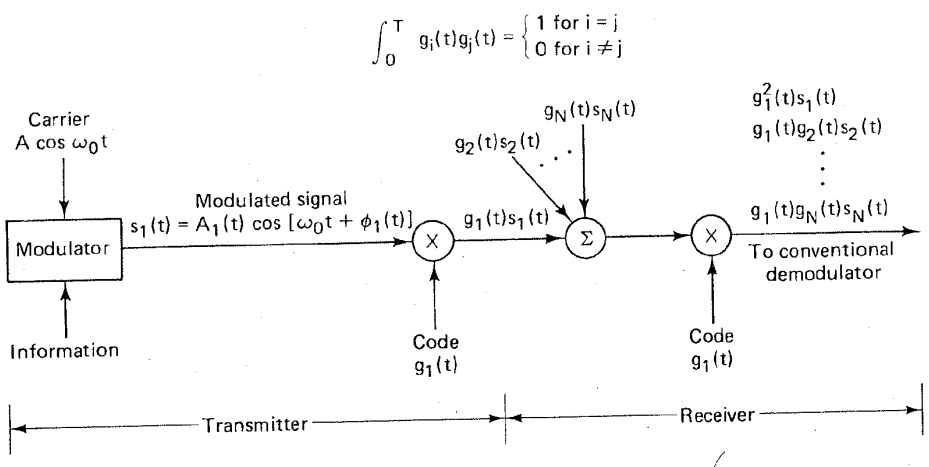


Figure 10.25 Code-division multiple access.

If the code functions, $\{g_i(t)\}$, are chosen with orthogonal properties, similar to Equation (10.14), the desired signal can be extracted perfectly in the absence of noise since $\int_0^T g_i^2(t) = 1$, and the undesired signals are easily rejected, since $\int_0^T g_i(t)g_j(t)\,dt = 0$ for $i \neq j$. In practice, the codes are not perfectly orthogonal; hence the cross-correlation between user codes introduces performance degradation, which limits the maximum number of simultaneous users.

Consider the frequency-domain view of the DS/CDMA receiver. Figure 10.26a illustrates the wideband input to the receiver; it consists of wanted and unwanted signals, each spread by its own code with code rate R_p, and each having a power spectral density of the form $\text{sinc}^2(f/R_p)$. Receiver thermal noise is also shown as having a flat spectrum across the band. The combined waveform of Equation (10.39) (desired plus undesired signals) is applied to the input of the receiver correlator driven by a synchronous replica of $g_1(t)$. Figure 10.26b illustrates the spectrum after correlation with the code $g_1(t)$ (despreading). The desired

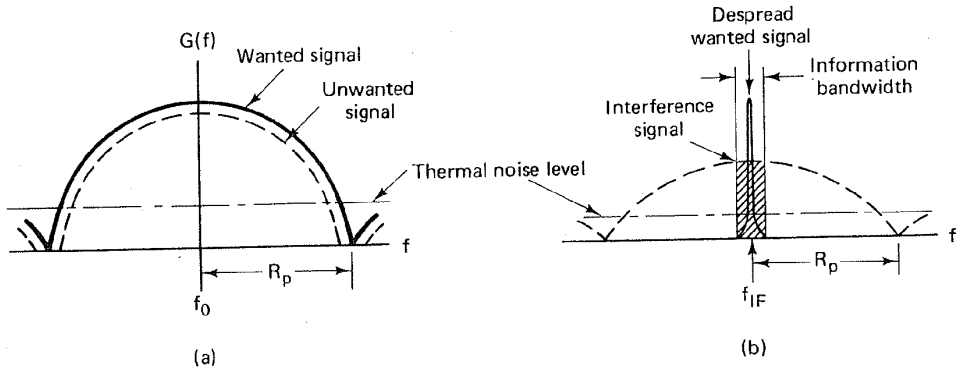


Figure 10.26 Spread-spectrum signal detection. (a) Spectrum at the input to receiver. (b) Spectrum after correlation with the correct and synchronized PN code.

signal, occupying the information bandwidth centered at an intermediate frequency (IF), is then applied to a conventional demodulator, with bandwidth just wide enough to accommodate the despread signal. The undesired signals of Equation (10.39) remain effectively spread by $g_1(t)g_i(t)$. Only that portion of the spectrum of the unwanted signals falling in the information bandwidth of the receiver will cause interference with the desired signal.

Pursley [17] presents an excellent treatment on the performance of SSMA using DS, taking correlation properties of the code sequences into account. Also, Geraniotis [18] and Geraniotis and Pursley [19, 20] evaluate the performance of FH and DS multiple access systems subject to interference.

10.6.2 Multipath Channels

Consider a DS binary PSK communication system operating over a multipath channel that has more than one path from the transmitter to the receiver. Such multiple paths may be due to atmospheric reflection or refraction, or reflections from buildings or other objects, and may result in fluctuations in the received signal level. The different paths may consist of several discrete paths each with a different attentuation and time delay, or they might consist of a continuum of paths. Figure 10.27 illustrates a communication link with two discrete paths. The multipath wave is delayed by some time, τ, compared to the direct wave. In television receivers, signals such as these cause "ghosts," or under extreme conditions, complete loss of picture synchronization.

In a direct-sequence spread-spectrum system, if we assume that the receiver is synchronized to the time delay and RF phase of the direct path, the received signal can be expressed as

$$r(t) = Ax(t)g(t)\cos\omega_0 t + \alpha Ax(t-\tau)g(t-\tau)\cos(\omega_0 t + \theta) + n(t) \quad (10.40)$$

where $x(t)$ is the data signal, $g(t)$ the code signal, $n(t)$ a zero-mean Gaussian noise process, and τ the differential time delay between the two paths, assumed to be in the interval $0 < \tau < T$. The angle θ is a random phase, assumed to be uniformly distributed in the range $(0, 2\pi)$, and α is the attenuation of the multipath signal

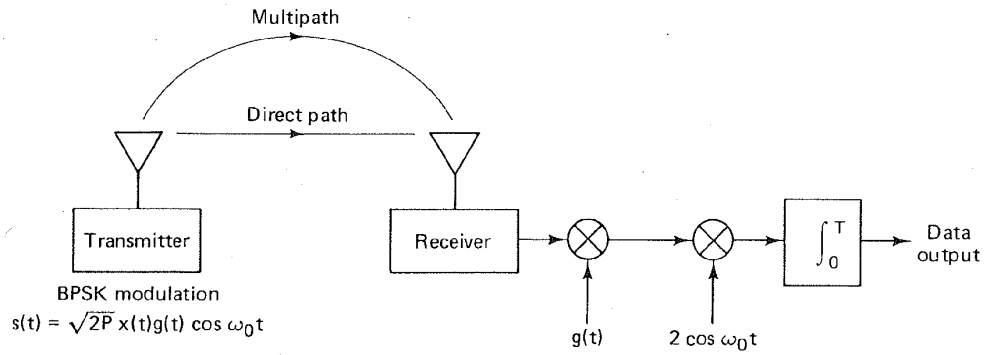


Figure 10.27 Direct-sequence BPSK system operating over a multipath channel.

relative to the direct path signal. For the receiver, synchronized to the direct path signal, the output of the correlator, $z(t = T)$, can be written as

$$z(t = T) = \int_0^T [Ax(t)g^2(t) \cos \omega_0 t + \alpha Ax(t - \tau)g(t)g(t - \tau) \cos (\omega_0 t + \theta) + n(t)g(t)]2 \cos \omega_0 t \, dt \quad (10.41)$$

where $g^2(t) = 1$. Also, for $\tau > T_c$, $g(t)g(t - \tau) \simeq 0$ (for codes with long periods), where T_c is the chip duration. Therefore, if T_c is less than the differential time delay between the multipath and direct path signals, we can write

$$z(t = T) = \int_0^T 2Ax(t)\cos^2 \omega_0 t + 2n(t)g(t)\cos \omega_0 t \, dt = Ax(T) + n_0(T) \quad (10.42)$$

where $n_0(T)$ is a zero-mean Gaussian random variable. We see that the spread-spectrum system, similar to the case of CDMA, effectively eliminates the multipath interference by virtue of its code-correlation receiver.

If frequency hopping (FH) is used against the multipath problem, improvement in system performance is also possible but through a different mechanism. FH receivers avoid multipath losses by rapid changes in the transmitter frequency band, thus avoiding the interference by changing the receiver band position before the arrival of the multipath signal.

10.6.3 The Jamming Game

The goals of a jammer are to deny reliable communications to his adversary and to accomplish this at minimum cost. The goals of the communicator are to develop a jam-resistant communication system under the following assumptions: (1) complete invulnerability is not possible; (2) the jammer has a priori knowledge of most system parameters, such as frequency bands, timing, traffic, and so on; (3) the jammer has *no* a priori knowledge of the PN spreading or hopping codes. The signaling waveform should be designed so that the jammer cannot gain any appreciable jamming advantage by choosing a jammer waveform and strategy other than wideband Gaussian noise (i.e., being clever should gain nothing for the jammer). The fundamental design rule in specifying a jam-resistant system is to make it as costly as possible for the jammer to succeed in jamming the system.

10.6.3.1 Jammer Waveforms

There are many different waveforms that can be used for jamming communication systems. The most appropriate choice depends on the targeted system. Figure 10.28 shows power spectral density plots of examples of jammer waveforms versus a communicator's frequency hopped M-ary FSK (FH/MFSK) tone. The range of the abscissa represents the spread-spectrum bandwidth W_{ss}. The three columns in the figure represent three instances in time (three hop times) when symbols having spectra G_1, G_2, and G_3, respectively, are being transmitted. Figure 10.28a illustrates a relatively low-level noise jammer occupying the full spread-spectrum bandwidth. In Figure 10.28b the jammer strategy is to trade bandwidth

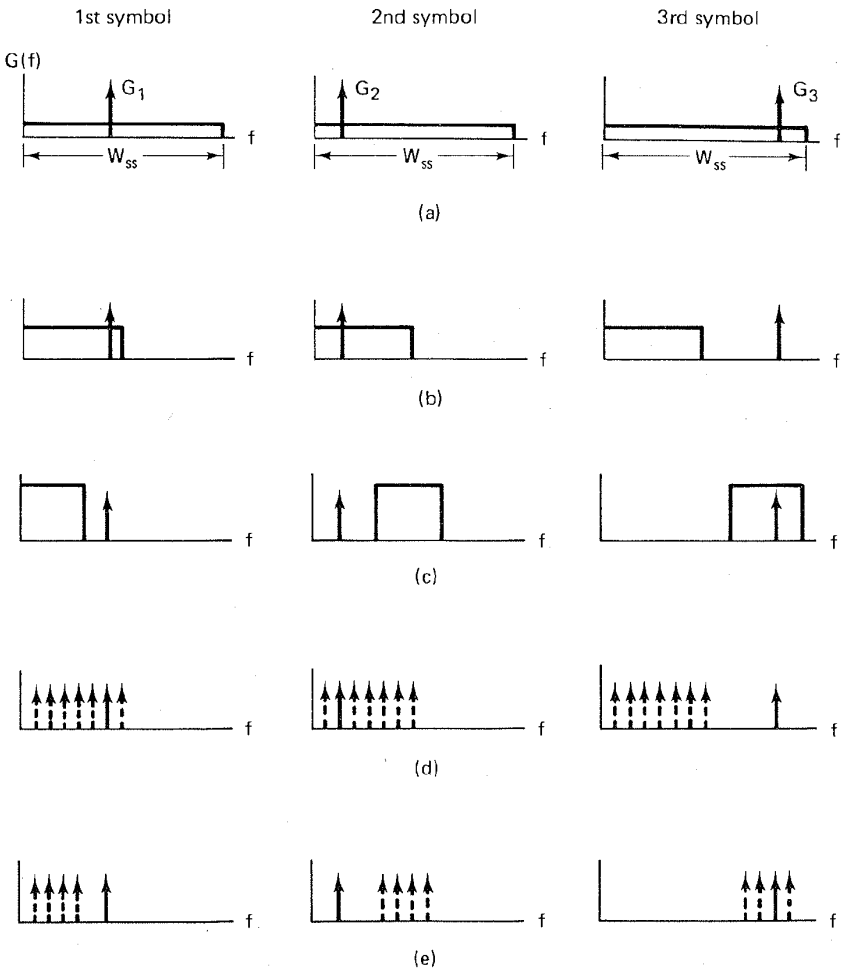


Figure 10.28 Jammer waveforms. (a) Full-band noise. (b) Partial-band noise. (c) Stepped noise. (d) Partial-band tones. (e) Stepped tones.

occupancy for greater power spectral density (the total power, or area under the curve, remains the same). The figure indicates that in this case, the jammer noise does not always share the same bandwidth region as the signal, but when it does, the effect can be destructive. In Figure 10.28c the noise jammer strategy is again to jam only part of the band, so that the jammer power spectral density can be increased, but in this case the jammer steps through different regions of the band at random times, thus preventing the communicator from using adaptive techniques to avoid the jamming. In Figure 10.28d and e the jammer uses a group of tones, instead of a continuous frequency band, in partial-band (Figure 10.28d) and stepped fashion (Figure 10.28e). This is a technique most often used against FH systems. Another jamming technique, not shown in Figure 10.28, is a pulse

jammer, consisting of pulse-modulated bandlimited noise. Unless otherwise stated, we shall assume that the jammer waveform is wideband noise and that the jammer strategy is to jam the entire bandwidth W_{ss} continuously. The effects of partial band jamming and pulse jamming are considered later.

10.6.3.2 Tools of the Communicator

The usual design goal for an anti-jam (AJ) communication system is to force a jammer to expend its resources over (1) a wide-frequency band, (2) for a maximum time, and (3) from a diversity of sites. The most prevalent design options are (1) frequency diversity, by the use of direct-sequence and frequency hopping spread-spectrum techniques; (2) time diversity, by the use of time hopping; (3) spatial discrimination, by the use of a narrow-beam antenna which forces a jammer to enter the receiver via an antenna sidelobe and hence suffer, typically, a 20- to 25-dB disadvantage, and (4) combinations of the above.

10.6.3.3 *J/S* Ratio

In Chapter 4 we were concerned primarily with link error performance as a function of thermal noise interference. Emphasis was placed on the signal-to-noise ratio parameters—required E_b/N_0 and available E_b/N_0 for meeting a specified error performance. In this section we are similarly concerned with link error performance as a function of interference. However, here the source of interference is the noise power of a jammer in addition to thermal noise. Therefore, the SNR of interest is $E_b/(N_0 + J_0)$, where J_0 is the noise power spectral density due to the jammer. Unless otherwise specified, J_0 is assumed equal to J/W_{ss}, where J is the average received jammer power (jammer power referred to the receiver front end) and W_{ss} is the spread-spectrum bandwidth. Since the jammer power is generally much greater than the thermal noise power, the SNR of interest in a jammed environment is usually taken to be E_b/J_0. Therefore, similar to the thermal noise case, we define $(E_b/J_0)_{\text{reqd}}$ as the bit energy per jammer noise power spectral density *required* for maintaining the link at a specified error probability. The parameter E_b can be written as

$$E_b = ST_b = \frac{S}{R}$$

where S is the received signal power, T_b the bit duration, and R the data rate in bits/s. Then we can express $(E_b/J_0)_{\text{reqd}}$ as

$$\left(\frac{E_b}{J_0}\right)_{\text{reqd}} = \left(\frac{S/R}{J/W_{ss}}\right)_{\text{reqd}} = \frac{W_{ss}/R}{(J/S)_{\text{reqd}}} = \frac{G_p}{(J/S)_{\text{reqd}}} \tag{10.43}$$

where $G_p = W_{ss}/R$ is denoted the *processing gain*, and $(J/S)_{\text{reqd}}$ can be written

$$\left(\frac{J}{S}\right)_{\text{reqd}} = \frac{G_p}{(E_b/J_0)_{\text{reqd}}} \tag{10.44}$$

The ratio $(J/S)_{\text{reqd}}$ is a figure of merit that provides a measure of how *invulnerable*

a system is to interference. Which system has better jammer-rejection capability: one with a larger $(J/S)_{\text{reqd}}$ or a smaller $(J/S)_{\text{reqd}}$? The *larger* the $(J/S)_{\text{reqd}}$, the *greater* is the system's noise rejection capability, since this figure of merit describes how much noise power relative to signal power is *required* in order to degrade the system's specified error performance. Of course, the communicator would like the communication system *not* to degrade at all.

Another way of describing the relationship in Equation (10.44) is as follows. An adversary would like to employ a jamming strategy that forces the effective $(E_b/J_0)_{\text{reqd}}$ to be as large as possible. The adversary may employ pulse, tone, or partial-band jamming rather than wideband noise jamming. A large $(E_b/J_0)_{\text{reqd}}$ implies a small $(J/S)_{\text{reqd}}$ ratio for a fixed processing gain. This may force the communicator to employ a larger processing gain to increase the $(J/S)_{\text{reqd}}$. The system designer strives to choose a signaling waveform such that the jammer can gain no special advantage by using a jamming strategy other than wideband Gaussian noise.

10.6.3.4 Anti-Jam Margin

Sometimes the $(J/S)_{\text{reqd}}$ ratio is referred to as the *anti-jam* (AJ) *margin* since it characterizes the system jammer-rejection capability. But this is not really a good use of the phrase since AJ margin usually means the safety margin against a *particular threat*. Using the same approach as in Chapter 4 (for calculating the margin against thermal noise), we can define the AJ margin M_{AJ}, as follows:

$$M_{\text{AJ}}(\text{dB}) = \left(\frac{E_b}{J_0}\right)_r (\text{dB}) - \left(\frac{E_b}{J_0}\right)_{\text{reqd}} (\text{dB}) \tag{10.45}$$

where $(E_b/J_0)_r$ is the E_b/J_0 *actually received*. Following the same format as Equation (10.43), we can express $(E_b/J_0)_r$ as

$$\left(\frac{E_b}{J_0}\right)_r = \frac{G_p}{(J/S)_r} \tag{10.46}$$

where $(J/S)_r$, or simply J/S, is the ratio of the actually received jammer power to signal power. We can now combine Equations (10.43), (10.45), and (10.46), as follows:

$$M_{\text{AJ}}\ (\text{dB}) = \frac{G_p}{(J/S)_r} (\text{dB}) - \frac{G_p}{(J/S)_{\text{reqd}}} (\text{dB}) \tag{10.47}$$

$$= \left(\frac{J}{S}\right)_{\text{reqd}} (\text{dB}) - \left(\frac{J}{S}\right)_r (\text{dB}) \tag{10.48}$$

Example 10.2 Satellite Jamming

Figure 10.29 illustrates a satellite jamming scenario. The airplane terminal is equipped with a frequency hopping (FH) spread-spectrum system transmitting with an EIRP_T = 20 dBW. The data rate is R = 100 bits/s. The jammer is transmitting wideband Gaussian noise, continually, with an EIRP_J = 60 dBW. Assume that $(E_b/J_0)_{\text{reqd}}$ = 10 dB and that the path loss is identical for both the airplane terminal and the jammer.

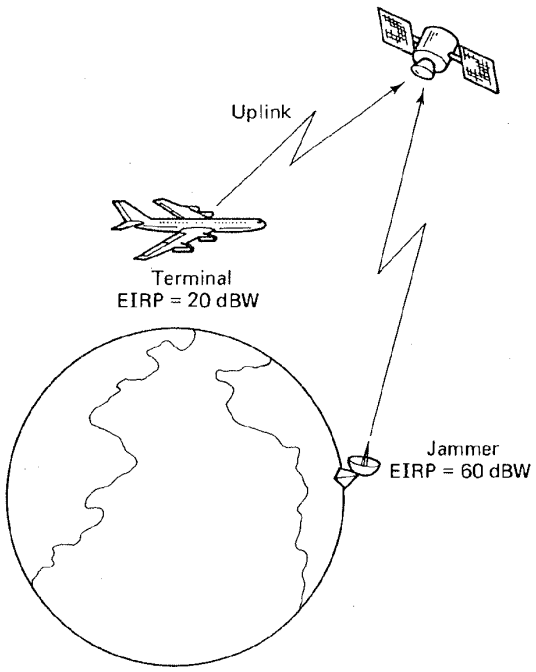


Figure 10.29 Satellite jamming scenario.

(a) Should the communicators be concerned more with the jamming of the uplink or with that of the downlink?
(b) If it is desired to have an AJ margin of 20 dB, what should be the value of the hopping bandwidth W_{ss}?

Solution

(a) Jamming the uplink is of much greater concern, since such single-point interference could degrade the communications of a multitude of terminals that are simultaneously using the satellite transponder. To achieve an equivalent degradation by jamming the downlink, the jammer would have to jam each of the receiving terminals. Downlink jamming is of some concern for critical military missions, but of less concern than uplink jamming.
(b) With the assumption that the path loss is the same for both the communicator and the jammer, we can replace $(J/S)_r$ in Equation (10.48) with the ratio of *transmitted* jammer-to-signal power, $\text{EIRP}_J/\text{EIRP}_T$. Therefore, we can write

$$M_{\text{AJ}}\,(\text{dB}) = (J/S)_{\text{reqd}}\,(\text{dB}) + \text{EIRP}_T\,(\text{dBW}) - \text{EIRP}_J\,(\text{dBW})$$

$$= G_p\,(\text{dB}) - \left(\frac{E_b}{J_0}\right)_{\text{reqd}}(\text{dB}) + \text{EIRP}_T\,(\text{dBW}) - \text{EIRP}_J\,(\text{dBW})$$

$$G_p = 20\text{ dB} + 10\text{ dB} - 20\text{ dBW} + 60\text{ dBW} = 70\text{ dB}$$

$$W_{ss} = G_p\,(\text{dB}) + R\,(\text{dB-Hz}) = 70\text{ dB} + 20\text{ dB-Hz}$$

$$= 90\text{ dB-Hz} = 1\text{ GHz}$$

Example 10.3 Satellite Downlink Jamming

In Example 10.2 the distance from the transmitting airplane to the receiving satellite and the distance from the jammer to the satellite were assumed identical. Certainly, the closer the jammer gets to the receiver, the greater will be the jamming interference. Consider a downlink jamming scenario where the satellite $\text{EIRP}_s = 35$ dBW, the jammer $\text{EIRP}_J = 60$ dBW, the space loss from the satellite to the receiving terminal is $L_s = 200$ dB, and the space loss from the jammer to the receiving terminal is $L_s' = 160$ dB. How much processing gain is needed to close the link with an AJ margin of 0 dB? Assume that $(E_b/J_0)_{\text{reqd}} = 10$ dB.

Solution

For the downlink jamming scenario the proximity of the jammer to the receiving airplane is much closer than that of the satellite to the airplane. These distances show up as the space losses in the $(J/S)_r$ term of Equation (10.48), as follows:

$$M_{\text{AJ}}\,(\text{dB}) = \left(\frac{J}{S}\right)_{\text{reqd}}(\text{dB}) - \left(\frac{J}{S}\right)_r(\text{dB})$$

where

$$\left(\frac{J}{S}\right)_r(\text{dB}) = \text{EIRP}_J\,(\text{dBW}) - L_s'\,(\text{dB}) - \text{EIRP}_s\,(\text{dBW}) + L_s\,(\text{dB})$$

and

$$\left(\frac{J}{S}\right)_{\text{reqd}}(\text{dB}) = \frac{W_{ss}}{R}\,(\text{dB}) - \left(\frac{E_b}{J_0}\right)_{\text{reqd}}(\text{dB})$$

Combining the above equations, and solving for processing gain, $G_p = W_{ss}/R$, yields

$$G_p\,(\text{dB}) = 75\ \text{dB}$$

10.7 FURTHER JAMMING CONSIDERATIONS

10.7.1 Broadband Noise Jamming

If the jamming signal is modeled as a zero-mean wide-sense-stationary Gaussian noise process with a flat power spectral density over the frequency range of interest, then for a fixed jammer received power, J, the jammer power spectral density J_0' is equal to J/W, where W is the bandwidth that the jammer chooses to occupy. If the jammer strategy is to jam the entire spread-spectrum bandwidth, W_{ss}, with its fixed power, the jammer is referred to as a wideband or *broadband jammer,* and the jammer power spectral density is

$$J_0 = \frac{J}{W_{ss}} \tag{10.49}$$

In Chapter 3 it was shown that the bit error probability P_B for a coherently

demodulated BPSK system (without channel coding) is

$$P_B = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \tag{10.50}$$

where $Q(x)$ is defined in Equations (2.42) and (2.43) and tabulated in Table B.1. The single-sided noise power spectral density N_0 represents thermal noise at the front end of the receiver. The presence of the jammer increases this noise power spectral density from N_0 to $(N_0 + J_0)$. Thus the average bit error probability for a coherent BPSK system in the presence of broadband jamming is

$$P_B = Q\left(\sqrt{\frac{2E_b}{N_0 + J_0}}\right) = Q\left[\sqrt{\frac{2E_b/N_0}{1 + (E_b/N_0)(J/S)/G_p}}\right] \tag{10.51}$$

When P_B is plotted versus E_b/N_0 for a given J/S ratio, the resulting curves are such as those in Figure 10.30, [7, 21]. The curves in Figure 10.30, shown for two different values of processing gain, *tend to flatten out* as E_b/N_0 increases, indicating that for a given ratio of jammer power to signal power, the jammer will cause some irreducible error probability. The only way to reduce this error probability is to increase the processing gain.

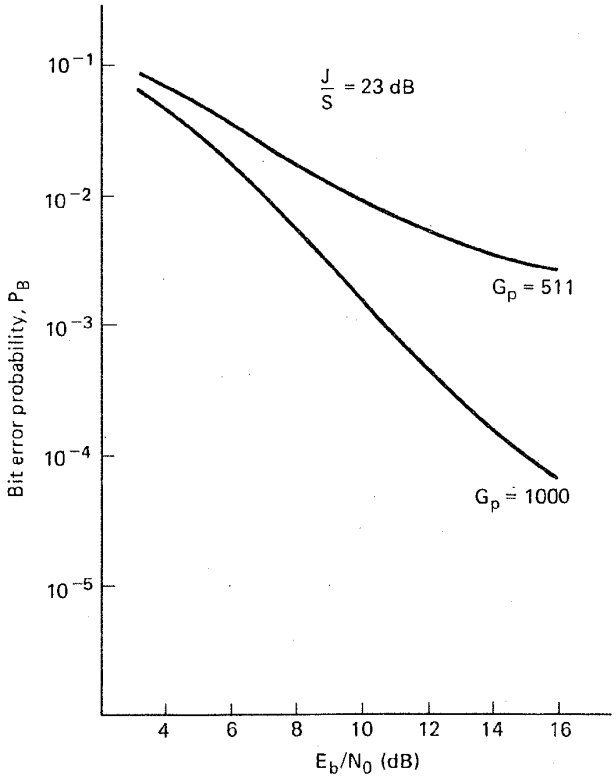


Figure 10.30 Bit error probability versus E_b/N_0 for a given J/S ratio. (Reprinted with permission from R. L. Pickholtz, D. L. Schilling, and L. B. Milstein, "Theory of Spread-Spectrum Communications—A Tutorial," *IEEE Trans. Commun.*, vol. COM30, no. 5, May 1982, Fig. 11, p. 866. © 1982 IEEE.)

10.7.2 Partial-Band Noise Jamming

A jammer can often increase the degradation to a FH system by employing *partial-band* jamming. Assuming that the frequency hopped modulation format is non-coherently detected binary FSK, the probability of a bit error, from Equation (3.111) is

$$P_B = \frac{1}{2} \exp\left(-\frac{E_b}{2N_0}\right) \tag{10.52}$$

Let us define a parameter, ρ, where $0 < \rho \leq 1$, representing the fraction of the band being jammed. The jammer can trade bandwidth jammed for in-band jammer power, such that by jamming a band $W = \rho W_{ss}$, the jammer noise power spectral density can be concentrated to a level J_0/ρ, thus maintaining a constant average jamming received power J where $J = J_0 W_{ss}$.

In the case of partial-band jamming, a specific transmitted symbol will be received unjammed, with probability $(1 - \rho)$, and will be perturbed by jammer power with spectral density J_0/ρ, with probability ρ. Therefore, the average bit error probability can be written from Equation (10.52), as follows:

$$P_B = \frac{1-\rho}{2} \exp\left(-\frac{E_b}{2N_0}\right) + \frac{\rho}{2} \exp\left[-\frac{E_b}{2(N_0 + J_0/\rho)}\right] \tag{10.53}$$

Since, in a jamming environment, it is often the case that $J_0 \gg N_0$, we can simplify Equation (10.53) to the form

$$P_B = \frac{\rho}{2} \exp\left(-\frac{\rho E_b}{2J_0}\right) \tag{10.54}$$

Figure 10.31 illustrates the probability of bit error versus E_b/J_0 for various values of the fraction, ρ. Clearly, the jammer would choose the fraction $\rho = \rho_0$ that maximizes P_B. Notice that ρ_0 decreases with increasing values of E_b/J_0 (see the ρ_0 locus in Figure 10.31). An expression for ρ_0 is easily found by differentiation (setting $dP_B/d\rho = 0$ and solving for ρ). This yields

$$\rho_0 = \begin{cases} \dfrac{2}{E_b/J_0} & \text{for } \dfrac{E_b}{J_0} > 2 \\ 1 & \text{for } \dfrac{E_b}{J_0} \leq 2 \end{cases} \tag{10.55}$$

In this case, $(P_B)_{\max}$ is given by

$$(P_B)_{\max} = \begin{cases} \dfrac{e^{-1}}{E_b/J_0} & \text{for } \dfrac{E_b}{J_0} > 2 \\ \dfrac{1}{2} \exp\left(-\dfrac{E_b}{2J_0}\right) & \text{for } \dfrac{E_b}{J_0} \leq 2 \end{cases} \tag{10.56}$$

where e is the base of the natural logarithm ($e = 2.7183$). This result is dramatic; the effect of a worst-case partial-band jammer on a system with spread spectrum *but without coding* changes the exponential relationship of Equation (10.54) into the inverse linear one of Equation (10.56). The ρ_0 locus in Figure 10.31 illustrates the P_B versus E_b/J_0 performance for the worst-case partial-band jammer. Here at 10^{-6} bit error probability there is over 40-dB difference between broadband noise jamming and the worst-case partial-band jamming for the same jamming

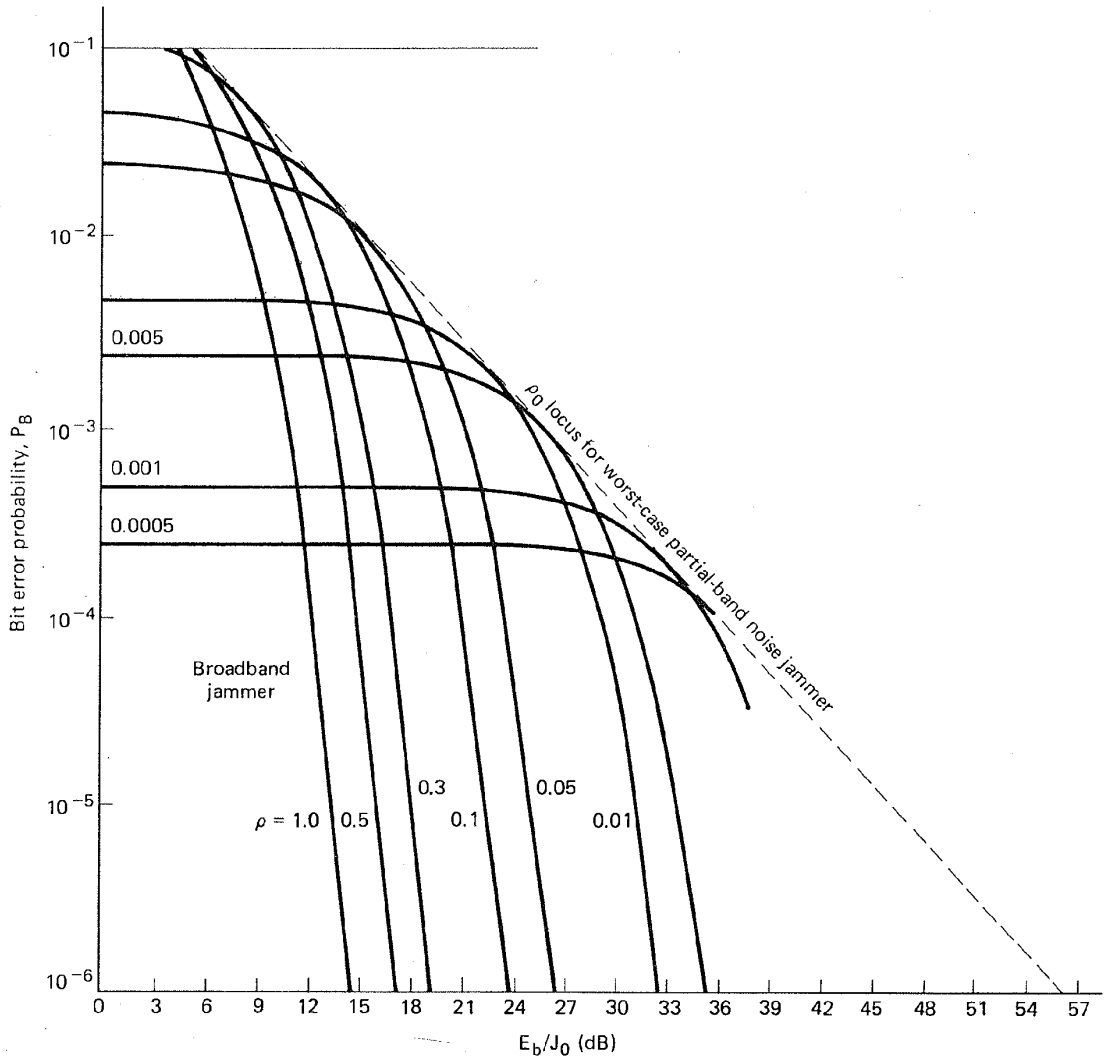


Figure 10.31 Partial-band noise jammer (FH/BFSK signaling). (Reprinted from M. K. Simon, J. K. Omura, R. A. Scholtz, and B. K. Levitt, *Spread Spectrum Communications*, Vol. 1, Fig. 3.24, p. 173. © 1985, with permission of the publisher, Computer Science Press, Inc., 1803 Research Blvd., Rockville, Md. 20850 USA.)

power [6, 22]. Hence, an intelligent jammer, with fixed finite power, can produce significantly greater degradation with partial-band jamming than is possible with broadband jamming. Forward error correction (FEC) coding with appropriate interleaving can mitigate this degradation [9]. In fact, for codes with low-enough rates, FEC can *force* a partial-band jammer to be a worst-case jammer only when operating as a broadband jammer [23, 24].

10.7.3 Multiple-Tone Jamming

In the case of *multiple-tone jamming,* the jammer divides its total received power, J, into distinct, equal-power, random-phase CW tones. These are distributed over the spread-spectrum bandwidth, W_{ss}, according to some strategy [9]. The analysis of the effects of tone jamming is more complicated than that of noise jamming, especially for DS systems. Therefore, the effect of a despread tone is often approximated as Gaussian noise. Reference [25] provides analysis of the performance of DS systems in the presence of multiple-tone interference. For a noncoherent FH/FSK system operating in the presence of partial-band tone jamming, the performance is often assumed the same as that of partial-band noise jamming [26]. However, multiple-CW-tone jamming can be more effective than partial-

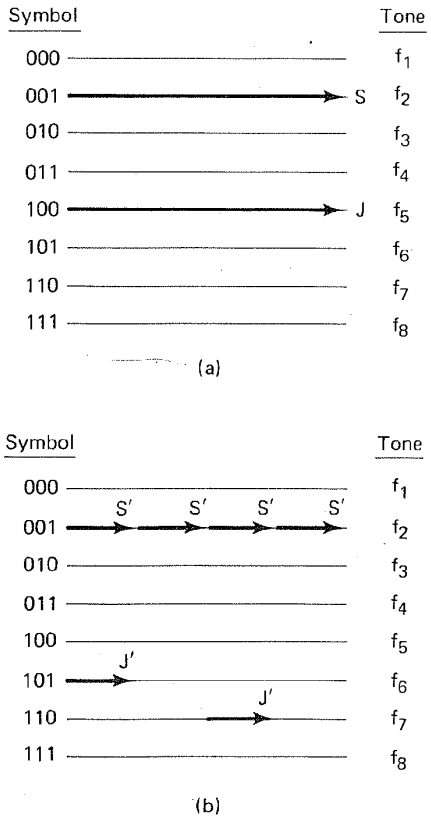


Figure 10.32 Fast hopping symbol repeat versus tone jamming. (a) One frequency hop. (b) Four frequency hops.

band noise against FH/MFSK signals because CW tones are the most efficient way for a jammer to inject energy into noncoherent detectors [9]. References [9, 10, 26, 27] provide extensive treatment and analysis of the performance of various communication systems in the presence of various types of jammers.

In the FFH/MFSK demodulator of Figure 10.16, a chip clipping circuit is shown between each envelope detector and accumulator. The function of such a circuit in a tone jamming environment can best be understood with the aid of the example shown in Figure 10.32. An 8-ary FSK frequency hopping system with no diversity, indicated in Figure 10.32a, is compared with a *fast*-frequency hopping system that combines chip repeating ($N = 4$ in this example) with the clipping of each chip, indicated in Figure 10.32b. Each row in the figures reprsents one of the $M = 8$ accumulators shown in Figure 10.16. The presence of a signal in the accumulator is indicated by a vector. In Figure 10.32a we see that, for a particular frequency hop, the data band is occupied by a received message symbol with received signal power, S. If, by chance, a jamming tone with received power J, where $J \geq S$, falls on a different tone within this data band during the same hop, the detector would not be able to decide reliably on the correct symbol.

In Figure 10.32b, the communicator's four chips (the length of each vector is a measure of the clipped signal power, S') sum to the maximum capacity of the accumulator. If the jammer tones, by chance, fall in the same spectral region as that of the signal, they will not confuse the detector, since the jamming tones are also clipped to the same level, $J' = S'$, as the signal chips. In Figure 10.32b, two of the jamming tones fall in the data band, but because they are clipped, there is no confusion about the correct symbol decision.

10.7.4 Pulse Jamming

Consider a spread-spectrum DS/BPSK communication system in the presence of a pulse-noise jammer. A pulse-noise jammer transmits pulses of bandlimited white Gaussian noise having a time-averaged received power, J, although the actual power during a jamming pulse duration is larger. Assume that the jammer can choose the center frequency and bandwidth of the noise to be the same as the receiver's center frequency and bandwidth. Assume also that the jammer can trade duty cycle for increased (concentrated) jammer power, such that if the jamming is present for a fraction $0 < \rho < 1$ of the time, then during this time, the jammer power spectral density is increased to a level J_0/ρ, thus maintaining a constant time-averaged power J (where $J = J_0 W_{ss}$ and W_{ss} is the system spread-spectrum bandwidth).

The bit error probability P_B for a coherently demodulated BPSK system (without channel coding) was given in Equation (10.50):

$$P_B = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

The single-sided noise power spectral density N_0 represents thermal noise at the front end of the receiver. The presence of the jammer increases this noise power

spectral density from N_0 to $(N_0 + J_0/\rho)$. Since the jammer transmits with duty cycle ρ, the average bit error probability is

$$P_B = (1 - \rho)Q\left(\sqrt{\frac{2E_b}{N_0}}\right) + \rho Q\left(\sqrt{\frac{2E_b}{N_0 + J_0/\rho}}\right) \tag{10.57}$$

We can generally assume that in a jamming environment, N_0 can be neglected. Therefore, we can write

$$P_B \simeq \rho Q\left(\sqrt{\frac{2E_b\rho}{J_0}}\right) \tag{10.58}$$

The jammer will, of course, attempt to choose the duty cycle ρ that maximizes P_B. Figure 10.33 illustrates P_B for various values of ρ. The value of $\rho = \rho_0$ that maximizes P_B decreases with increasing values of E_b/J_0, as was the case with partial-band jamming. This is seen by differentiating Equation (10.58) to obtain [6]

$$\rho_0 = \begin{cases} \dfrac{0.709}{E_b/J_0} & \text{for } \dfrac{E_b}{J_0} > 0.709 \\ 1 & \text{for } \dfrac{E_b}{J_0} \leq 0.709 \end{cases} \tag{10.59}$$

which results in the maximum bit error probability

$$(P_B)_{\max} = \begin{cases} \dfrac{0.083}{E_b/J_0} & \text{for } \dfrac{E_b}{J_0} > 0.709 \\ Q\left(\sqrt{\dfrac{2E_b}{J_0}}\right) & \text{for } \dfrac{E_b}{J_0} \leq 0.709 \end{cases} \tag{10.60}$$

The effect of a worst-case pulse jammer upon a system with spread spectrum *but without coding* changes the complementary error function relationship of Equation (10.58) into the inverse linear one of Equation (10.60). As a result, at an error probability of 10^{-6}, there is almost a 40-dB difference in E_b/J_0 between the broadband jammer and the worst-case pulse jammer (see Figure 10.33). For the same jammer power, the jammer can do considerably more harm to an uncoded DS/BPSK system with pulse jamming than with constant power jamming. The effect of a pulse-noise jammer on uncoded DS/BPSK is similar to the effect of a partial-band noise jammer on uncoded FH/BFSK, treated in Section 10.7.2. In both cases considerable degradation is brought about by concentrating more jammer power on a fraction of the transmitted uncoded symbols. Forward error correction coding with appropriate interleaving can almost fully restore this degraded performance [9, 23–25, 28].

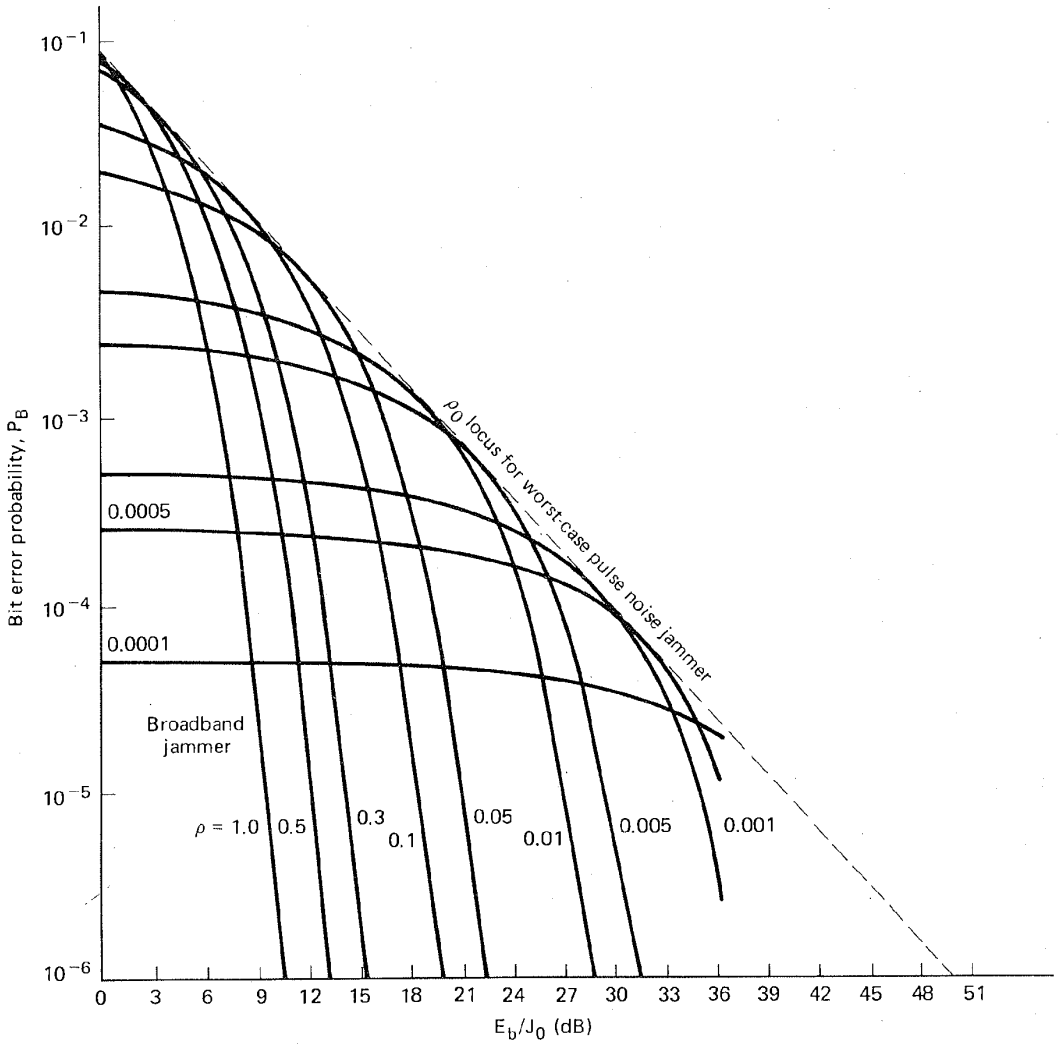


Figure 10.33 Pulse noise jammer (DS/BPSK signaling). (Reprinted from M. K. Simon, J. K. Omura, R. A. Scholtz, and B. K. Levitt, *Spread Spectrum Communications*, Vol. 1, Fig. 3.7, p. 150. © 1985, with permission of the publisher, Computer Science Press, Inc., 1803 Research Blvd., Rockville, Md. 20850 USA.)

10.7.5 Repeat-Back Jamming

In Examples 10.2 and 10.3 we considered an FH spread-spectrum system performance against a broadband Gaussian noise jammer. Notice that the frequency hopping rate did not enter into the margin computations. Isn't this disturbing? Intuitively, it would seem that the faster the frequency hops, the easier it is to "hide" the signal from the jammer. If the hopping rate truly does not enter into the computations, why not hop only once a day or once a week? The answer is that the meaure of jammer-rejection capability, namely processing gain, G_p, is based on the assumption that the jammer is a "dumb" jammer; that is, the jammer

knows the extent of the spread-spectrum bandwidth, W_{ss}, but does *not* know the exact spectral location of the signal at any moment in time. We assume that the hopping rate is *fast enough* to preclude the jammer from monitoring the transmitted signal so as to usefully change this jamming strategy. Under what condition is this assumption questionable? There are "smart" jammers that are known as *repeat-back jammers* or *frequency-follower (FF) jammers*. These jammers monitor a communicator's signal (usually via a sidelobe beam from the transmitting antenna). They possess wideband receivers and high-speed signal processing capability that enable them to rapidly concentrate their jamming signal power in the spectral vicinity of a communicator's FH/FSK signal. By so doing, the smart jammer can increase the jamming power in the communicator's instantaneous bandwidth, thereby gaining an advantage over a wideband jammer. Notice that this strategy is useful only against frequency hopping signals. In direct-sequence systems, there is no instantaneous narrowband signal for the jammer to detect.

What can be done to defeat the repeat-back jammer? One method is to simply hop so fast that by the time the jammer receives, detects, and transmits the jamming signal, the communicator is already transmitting at a *new* hop (which of course will be unaffected by jamming at the frequency of the prior hop). The following example should make this point clear.

Example 10.4 Fast Hopping to Evade the Repeat-Back Jammer

Assume that a repeat-back jammer is located $d = 30$ km away from the communicator. Assume further that the jammer can monitor any uplink transmission from the communicator to a nearby satellite, as shown in Figure 10.34. How fast must the communicator hop his frequency to evade the repeat-back jammer? Assume that the jammer can change its jamming frequency in zero time, and that the only differential delay between the communicator's uplink signal and the jamming uplink signal is the propagation delay from the communicator to the jammer.

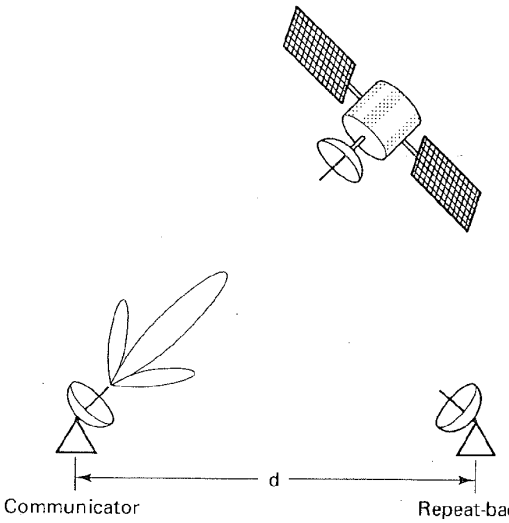


Figure 10.34 Example of fast hopping to evade the repeat-back jammer.

Solution

To ensure that the communicator's tone transmission and the jammer's attempt to disrupt that tone do not overlap in time, it is necessary that the duration of each hop, T_{hop}, have the value

$$T_{\text{hop}} \leq \frac{d}{c} = \frac{3 \times 10^4 \text{ m}}{3 \times 10^8 \text{ m/s}} = 10^{-4} \text{ s}$$

where c is the speed of light. Then $R_{\text{hop}} \geq 10{,}000$ hops/s.

10.7.6 BLADES System

Another technique capable of defeating the repeat-back jammer dates back to the mid-1950s when Sylvania engineers developed a system named the Buffalo Laboratories Application of Digitally Exact Spectra, or BLADES. The system used its code generator to independently select two new frequencies for each bit; the *final choice* of the frequency tone actually transmitted was dictated by the data bit about to be transmitted. Figure 10.35 illustrates a typical data stream of binary ones and zeros, called *marks* and *spaces*, respectively, and a sequence of frequency pairs f_1 and f_1', f_2 and f_2', ... The appearance of a mark dictates the choice of frequency f_i, while the appearance of a space dictates the choice of frequency f_i'. As shown in the figure, the data stream in this example gives rise to the sequence of transmitted tones, f_1', f_2, f_3', f_4', f_5, and so on. How can such a system defeat a repeat-back jammer? The jammer monitors the transmissions and sends up energy in the neighborhood of the frequencies it perceives. The modulation of the BLADES system has no structure in the usual sense; either there *is* energy present or there is *no* energy present at a given frequency. The jammer sending narrowband energy in the same spectral neighborhood as the signal, is not destroying any modulation structure. For a noncoherent system, the jammer is only enhancing the communicator's signal. The only recourse for the repeat-back jammer is to change strategy by becoming a broadband jammer, and to jam the entire spread-spectrum bandwidth.

Notice that it is not really necessary to have a *pair* of frequencies for each

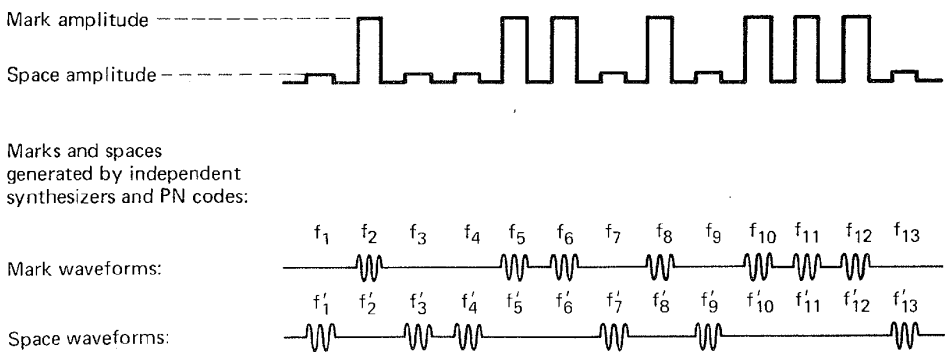


Figure 10.35 BLADES system.

bit. A *single* frequency will do. The communicator then transmits the pseudo-random frequency for a binary one and sends nothing for a binary zero. The receiver has the same code generator and therefore monitors the same pscudo-random frequencies. A binary one is detected by virtue of energy at the monitored frequency, and a binary zero is known by a lack of energy at the monitored frequency. Of course, the system is not as robust as when the marks and spaces are each transmitted on independently selected frequencies.

10.8 CONCLUSION

Spread-spectrum (SS) technology has only emerged since the 1950s. Yet this novel approach to applications such as multiple access, ranging, and interference rejection has rendered SS techniques extremely important to most current NASA and military communication systems. In this chapter we presented an overview enumerating the benefits and types of spread-spectrum techniques, as well as some historical background.

Pseudorandom sequences are at the heart of all present-day SS systems; we therefore treated PN generation and properties. Emphasis was placed on the two major spread-spectrum techniques: direct sequence and frequency hopping. Also, consideration was given to synchronization, a crucial aspect of spread-spectrum operation. Some application examples were considered, such as code-division multiple access and communications with multipath conditions. Also, attention was devoted to the subject of jamming and jam-resistant systems, since this area represents one of the primary uses for spread-spectrum systems.

REFERENCES

1. Scholtz, R. A., "The Origins of Spread Spectrum Communications," *IEEE Trans. Commun.*, vol. COM30, no. 5, May 1982, pp. 822–854.
2. Shannon, C. E., "Communication in the Presence of Noise," *Proc. IRE,* Jan. 1949, pp. 10–21.
3. Dillard, R. A., "Detectability of Spread Spectrum Signals," *IEEE Trans. Aerosp. Electron. Syst.,* July 1979.
4. Simon, M. K., Omura, J. K., Scholtz, R. A., and Levitt, B. K., *Spread Spectrum Communications,* Vol. 3, Computer Science Press, Inc., Rockville, Md., 1985.
5. de Rosa, L. A., and Rogoff, M., Sec. I (Communications) of *Application of Statistical Methods to Secrecy Communication Systems,* Proposal 946, Fed. Telecommun. Lab., Nutley, N.J., Aug. 28, 1950.
6. Simon, M. K., Omura, J. K., Scholtz, R. A., and Levitt, B. K., *Spread Spectrum Communications,* Vol. 1, Computer Science Press, Inc., Rockville, Md., 1985.
7. Pickholtz, R. L., Schilling, D. L., and Milstein, L. B., "Theory of Spread-Spectrum Communications—A Tutorial," *IEEE Trans. Commun.,* vol. COM30, no. 5, May 1982, pp. 855–884.

8. Pickholtz, R. L., Schilling, D. L., and Milstein, L. B., Revisions to "Theory of Spread-Spectrum Communications—A Tutorial," *IEEE Trans. Commun.*, vol. COM32, no. 2, Feb. 1984, pp. 211–212.
9. Simon, M. K., Omura, J. K., Scholtz, R. A., and Levitt, B. K., *Spread Spectrum Communications,* Vol. 2, Computer Science Press, Inc., Rockville, Md., 1985.
10. Simon, M. K., and Polydoros, A., "Coherent Detection of Frequency-Hopped Quadrature Modulations in the Presence of Jamming: Part I. QPSK and QASK; Part II. QPR class I Modulation," *IEEE Trans. Commun.*, vol. COM29, Nov. 1981, pp. 1644–1668.
11. Ward, R. B., "Acquisition of Pseudonoise Signals by Sequential Estimation," *IEEE Trans. Commun.*, COM13, Dec. 1965, pp. 475–483.
12. Spilker, J. J., and Magill, D. T., "The Delay-Lock Discriminator—An Optimum Tracking Device," *Proc. IRE,* Sept. 1961.
13. Spilker, J. J., "Delay-Lock Tracking of Binary Signals," *IEEE Trans. Space Electron. Telem.*, Mar. 1963.
14. Simon, M. K., "Noncoherent Psuedonoise Code Tracking Performance of Spread Spectrum Receivers," *Commun.,* vol. COM25, Mar. 1977.
15. Ziemer, R. E., and Peterson, R. L., *Digital Communications and Spread Spectrum Systems,* Macmillan Publishing Company, New York, 1985.
16. Holmes, J. K., *Coherent Spread Spectrum Systems,* John Wiley & Sons, Inc., New York, 1982.
17. Pursley, M. B., "Performance Evaluation for Phase-Coded Spread-Spectrum Multiple-Access Communication: Part I. System Analysis," *IEEE Trans. Commun.*, vol. COM25, no. 8, Aug. 1977, pp. 795–799.
18. Geraniotis, E., "Noncoherent Hybrid DS-SFH Spread-Spectrum Multiple-Access Communications," *IEEE Trans. Commun.*, vol. COM34, no. 9, Sept. 1986, pp. 862–872.
19. Geraniotis, E., and Pursley, M. B., "Error Probability for Direct-Sequence Spread-Spectrum Multiple-Access Communications: Part I. Upper and Lower Bounds," *IEEE Trans. Commun.*, vol. COM30, no. 5, May 1982, pp. 985–995.
20. Geraniotis, E., and Pursley, M. B., "Error Probabilities for Direct-Sequence Spread-Spectrum Multiple-Access Communications: Part II. Approximations," *IEEE Trans. Commun.*, vol. COM30, no. 5, May 1982, pp. 996–1009.
21. Schilling, D. L., Milstein, L. B., Pickholtz, R. L., and Brown, R. W., "Optimization of the Processing Gain of an *M*-ary Direct Sequence Spread Spectrum Communication System," *IEEE Trans. Commun.*, vol. COM28, no. 8, Aug. 1980, pp. 1389–1398.
22. Viterbi, A. J., and Jacobs, I. M., "Advances in Coding and Modulation for Noncoherent Channels Affected by Fading, Partial Band, and Multiple Access Interference," in A. S. Viterbi, ed., *Advances in Communication Systems,* Vol. 4, Academic Press, Inc., New York, 1975.
23. Stark, W. E., "Coding for Frequency-Hopped Spread-Spectrum Communication with Partial-Band Interference: Part I. Capacity and Cutoff Rate," *IEEE Trans. Commun.*, vol. COM33, no. 10, Oct. 1985, pp. 1036 – 1044.
24. Stark, W. E., "Coding for Frequency-Hopped Spread-Spectrum Communication with Partial-Band Interference: Part II. Coded Performance," *IEEE Trans. Commun.*, vol. COM33, no. 10, Oct. 1985, pp. 1045–1057.
25. Milstein, L. B., Davidovici, S., and Schilling, D. L., "The Effect of Multiple-Tone

Interfering Signals on a Direct Sequence Spread Spectrum Communication System," *IEEE Trans. Commun.*, vol. COM30, Mar. 1982, pp. 436–446.

26. Milstein, L. B., Pickholtz, R. L., and Schilling, D. L., "Optimization of the Processing Gain of an FSK-FH system," *IEEE Trans. Commun.*, vol. COM28, July 1980, pp. 1062–1079.
27. Huth, G. K., "Optimization of Coded Spread Spectrum Systems Performance," *IEEE Trans. Commun.*, vol. COM25, Aug. 1977, pp. 763–770.
28. Viterbi, A. J., "Spread Spectrum Communications—Myths and Realities," *IEEE Commun. Mag.*, May 1979, pp. 11–18.

PROBLEMS

10.1. Explain why a maximal-length n-stage linear feedback shift register can produce a sequence with a period no greater than $2^n - 1$.

10.2. Show that in a maximal-length n-stage linear feedback shift register the output stage must always be an input to the feedback network.

10.3. Consider the DS/BPSK spread-spectrum transmitter of Figure 10.9a or b. Let $x(t)$ be the sequence 1 0 0 1 1 0 0 0 1, arriving at a rate of 75 bits/s, where the leftmost bit is the earliest bit. Let $g(t)$ be generated by the shift register of Figure 10.7, with an initial state of 1 1 1 1 and a clock rate of 225 Hz.
- **(a)** Sketch the final transmitted sequence $x(t)g(t)$.
- **(b)** What is the bandwidth of the transmitted (spread) signal?
- **(c)** What is the processing gain?
- **(d)** Suppose that the estimated delay, $\hat{T}_d$, of Figure 10.9c is too large by one chip time. Sketch the despread chip sequence.
- **(e)** Choose a decision rule for deciding on $\hat{x}(t)$ and identify the errors.

10.4. A total of 24 equal-power terminals are to share a frequency band through a code-division multiple access (CDMA) system. Each terminal transmits information at 9.6 kbits/s with a direct-sequence spread-spectrum BPSK modulated signal. Calculate the minimum chip rate of the PN code in order to maintain a bit error probability of 10^{-3}. Assume that the receiver noise is negligible with respect to the interference from the other users.

10.5. A feedback shift register PN generator produces a 31-bit PN sequence at a clock rate of 10 MHz. What are the equation and graphical form of the autocorrelation function and power spectral density of the sequence? Assume that the pulses have values of ± 1.

10.6. Consider an FH/MFSK system such as the one shown in Figure 10.11. Let the PN generator be defined by a 20-stage linear feedback shift register with a maximal length sequence. Each state of the register dictates a new center frequency within the hopping band. The minimum step size between center frequencies (hop to hop) is 200 Hz. The register clock rate is 2 kHz. Assume that 8-ary FSK modulation is used and that the data rate is 1.2 kbits/s.
- **(a)** What is the hopping bandwidth?
- **(b)** What is the chip rate?
- **(c)** How many chips are there in each data symbol?
- **(d)** What is the processing gain?

10.7. The block diagram of Figure 10.16 is described in Section 10.4.5 for a fast frequency hopping (FFH) demodulator. Draw a similar block diagram for a slow frequency hopping (SFH) demodulator, and explain how it would work.

10.8. Find the mean and the standard deviation of the time needed to acquire a 10-megachip/s BPSK modulated PN code sequence using a serial search where 100 chips are examined at a time. Assume that a correct detection results when all 100 received chips match the locally generated ones. The ratio of received chip energy to noise power spectral density is 9.6 dB, and the uncertainty time between the received and local code sequences is 1 ms. Assume that the probability of false lock (false alarm) is negligible.

10.9. There are 11 equal-power terminals in a CDMA communication system, transmitting signals toward a central node. Each terminal transmits information at 1 kbit/s on a 100-kbits/s direct-sequence spreading signal using BPSK modulation.

(a) If receiver noise is negligible with respect to the interference from other users, what is the received ratio of bit energy to interference power spectral density (E_b/I_0) experienced by each user?

(b) What is the effect on E_b/I_0 if all users double their output power?

(c) If the users wish to expand their service to 101 equal-power users, what must be done to the spreading codes to maintain the original E_b/I_0 ratio?

10.10. A CDMA system uses direct-sequence modulation with a data bandwidth of 10 kHz and a spread bandwidth of 10 MHz. With only one signal being transmitted, the received E_b/N_0 is 16 dB.

(a) If the required E_b/N_0 is 10 dB, how many equal-power users can share the band?

(b) If each user's transmitted power is reduced by 3 dB, how many equal-power users can share the band?

(c) What is the maximum number of users that can share the band?

(d) How many equal-power users could share the band if they switch to TDMA with 98% efficient use of the communications resource?

(e) Why is the answer to part (d) so much greater than the answer to part (c)? What is the disadvantage of TDMA compared to CDMA (i.e., what penalty is paid to accommodate more users)?

10.11. A DS/SS system is used to combat multipath. If the path length of the multipath wave is 100 m longer than that of the direct wave, what is the minimum chip rate necessary to reject the multipath interference?

10.12. A ground-to-synchronous satellite link must be closed in a jamming environment. The data rate is 1 kbit/s and the ground station has a 60-ft antenna. Antijam protection is provided by a 10-Mbits/s direct-sequence spread-spectrum code. The jammer has a 150-ft antenna and a transmitter with 400 kW of power. Assume equal space and propagation losses. How much power is required of the earth station transmitter to achieve an E_b/J_0 of 16 dB at the satellite receiver? Assume that the receiver noise is negligible.

10.13. Input data at 75 bits/s are channel encoded using a rate $\frac{1}{2}$ encoder. The coded bits are then modulated using 8-ary FSK. The FSK symbols are then spread by frequency hopping at a rate of 2000 hops/s.

(a) What is the chip rate?

(b) What is the order of diversity?

(c) If there are two such signals, time-division multiplexed (TDM'd) on the channel

at the same hopping rate, how would this affect the chip rate, symbol rate, and order of diversity?

(d) If there are 80 such signals TDM'd on the channel, how would this effect the chip rate, symbol rate, and order of diversity?

10.14. A frequency hopping noncoherent binary FSK system operates at an E_b/N_0 of 30 dB with a hopping bandwidth of 2 GHz. Assume that no channel coding is used. A jammer operating over the same broadband bandwidth yields a received $J_0 = 100N_0$.

(a) What is the bit error probability, P_B?

(b) If the jammer becomes a partial-band jammer, what bandwidth should it occupy to be most effective?

(c) What is P_B as a result of such optimum partial-band jamming?

(d) What is the unjammed P_B?

10.15. A noncoherent frequency hopping 8-ary FSK system hops at 12,000 hops/s over a bandwidth of 1 MHz. The symbol rate is 3000 symbols/s. Assume that channel coding is not used. The signal power at the input of the receiver is 10^{-12} W. A partial-band noise jammer occupies 50 kHz (assumed to be entirely within the hopping bandwidth of the signal). The received jammer power is 10^{-11} W. Assume that the system temperature is 290 K. What is the probability of bit error?

10.16. A coherent DS/BPSK system is transmitting at a data rate of 10 kbits/s in the presence of a broadband jammer. Assume that the system does not use channel coding. Also assume that the propagation losses are the same for the system and the jammer.

(a) If the EIRP of the communicator is 20 kW and the EIRP of the jammer is 60 kW, calculate the required spread-spectrum bandwidth to achieve a bit error probability of $P_B = 10^{-5}$.

(b) If the jammer is a pulse jammer, calculate the pulse duty cycle that results in worst-case jamming. What is the value of P_B at this duty cycle?

10.17. A communicator intends to use frequency hopping at a hop rate of 10,000 hops/s to avoid a threat of repeat-back jamming.

(a) Ignoring the curvature of the earth, and assuming that the communicator is transmitting to a satellite at geosynchronous altitude (approximately 36,000 km) that is directly overhead, compute the *radius of vulnerability*, which is the radius outside of which the communicator is unconditionally safe from repeat-back jamming by a ground-based jammer.

(b) If the communicator knows that the jammer requires a minimum of 10 μs to identify the transmission frequency and tune the jammer output, compute the radius of vulnerability conditioned on this information.

10.18. Consider an airborne repeat-back jammer as shown in Figure P10.1. The communicator is using a FH/SS system. What is the minimum hop rate required in order that the repeat-back jamming does not degrade the message? What would be the minimum required hopping rate if the communicator and jammer switched positions (i.e., fixed land jammer and airborne communicator).

10.19. Spread-spectrum techniques can be used to meet government regulations regarding flux (power) density radiating the surface of the earth. If a satellite at synchronous altitude (36,000 km) transmits 4-kbits/s data using 100 W of EIRP, what spreading bandwidth is required to maintain a flux density on the earth's surface no greater than -151 dBW/m^2 in any 4-kHz band?

10.20. A communicator uses noncoherent BFSK modulation and frequency hopping to

combat the effects of a jammer. The power of the communicator's signal at the receiver input is 10 μW. The SNR in the absence of jamming is assumed to be very large. The power of the jamming signal at the receiver input is 1 W.

(a) If the jammer jams the entire hopping bandwidth with equal amounts of Gaussian noise (the noise will be white within the band), what bandwidth expansion factor will allow the communicator to maintain a bit error probability of 10^{-4}?

(b) Assume that the jammer decides to "color" its jamming noise by reducing its energy by a fraction, α ($0 \leq \alpha \leq 1$), in half the hop bandwidth, and increasing it by a like amount in the other half (thereby keeping its transmitted energy constant). Assuming that the communicator does not modify his hopping pattern to avoid the jammer strategy, develop an expression for the bit error probability for this case of colored jamming.

(c) Determine the fraction, α, that is optimum from the jammer standpoint for each of the limiting cases (i) when the effective SNR is large and, (ii) when it is small.

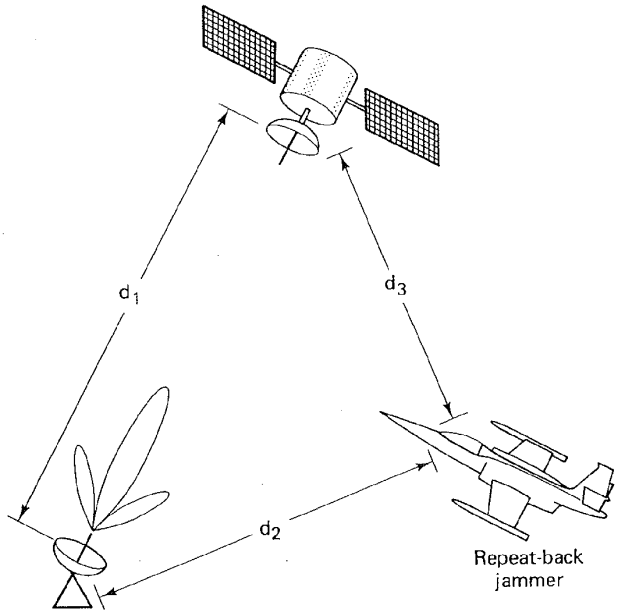
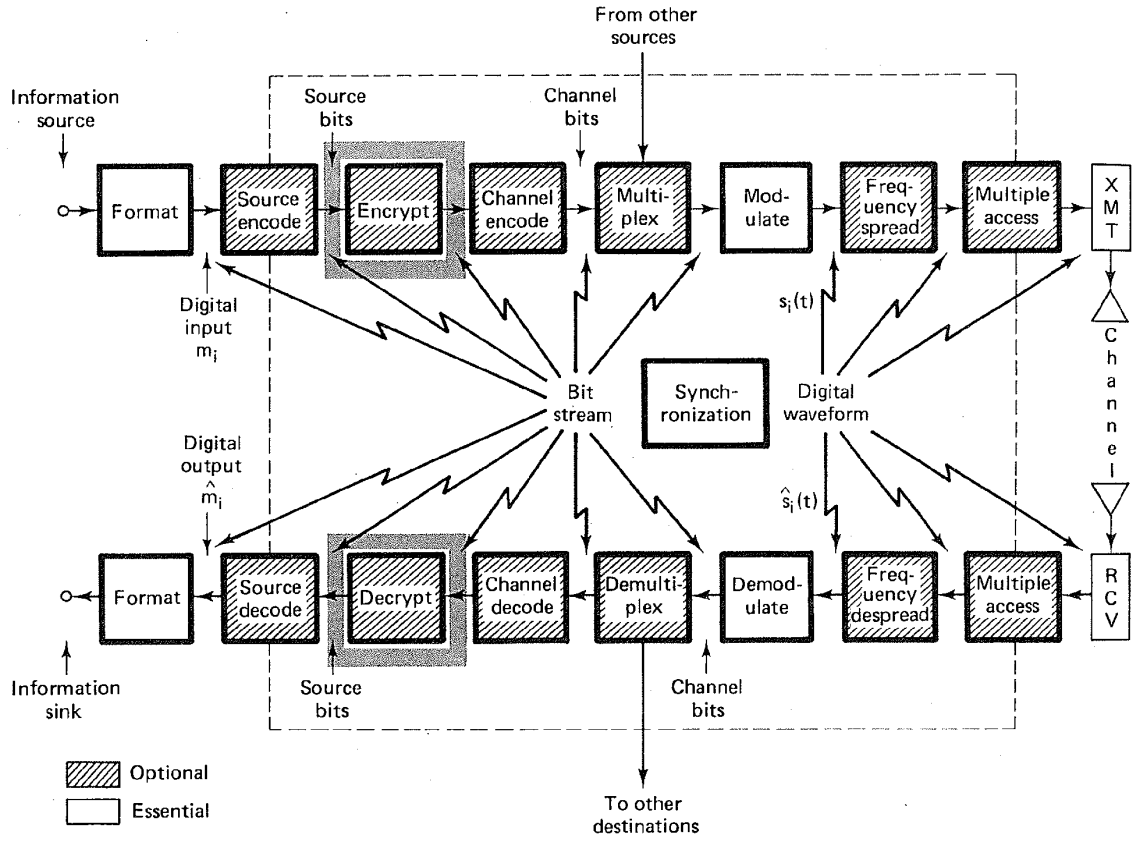


Figure P10.1

CHAPTER 12

Encryption and Decryption



12.1 MODELS, GOALS, AND EARLY CIPHER SYSTEMS

12.1.1 A Model of the Encryption and Decryption Process

The desire to communicate privately is a human trait that dates back to earliest times. Hence the history of secret communications is rich with unique inventions and colorful anecdotes [1]. The study of ways to disguise messages so as to avert unauthorized interception is called *cryptography*. The terms *encipher* and *encrypt* refer to the message transformation performed at the transmitter, and the terms *decipher* and *decrypt* refer to the inverse transformation performed at the receiver. The two primary reasons for using cryptosystems in communications are (1) *privacy*, to prevent unauthorized persons from extracting information from the channel (eavesdropping); and (2) *authentication*, to prevent unauthorized persons from injecting information into the channel (spoofing). Sometimes, as in the case of electronic funds transfer or contract negotiations, it is important to provide the electronic equivalent of a *written signature* in order to avoid or settle any dispute between the sender and receiver as to what message, if any, was sent.

Figure 12.1 illustrates a model of a cryptographic channel. A message, or plaintext, M, is encrypted by the use of an invertible transformation, E_K, that produces a ciphertext, $C = E_K(M)$. The ciphertext is transmitted over an insecure or *public channel*. When an authorized receiver obtains C, he decrypts it with the inverse transformation, $D_K = E_K^{-1}$, to obtain the original plaintext message, as follows:

$$D_K(C) = E_K^{-1}[E_K(M)] = M \tag{12.1}$$

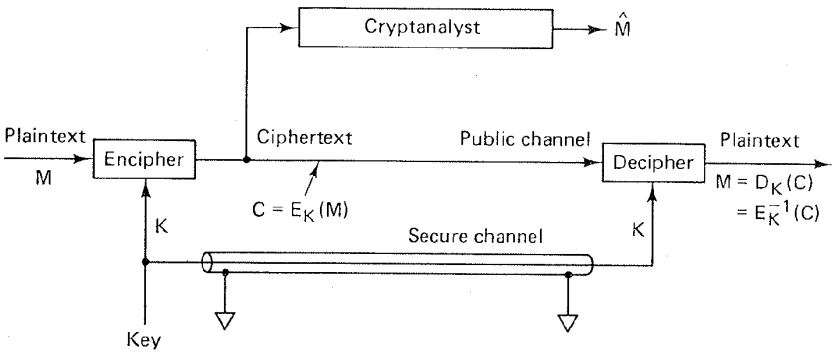


Figure 12.1 Model of a cryptographic channel.

The parameter K refers to a set of symbols or characters called a *key*, which dictates a specific encryption transformation, E_K, from a family of cryptographic transformations. Originally, the security of cryptosystems depended on the secrecy of the entire encryption process, but eventually systems were developed for which the general nature of the encryption transformation or algorithm could be publicly revealed, since the security of the system depended on the specific key. The key is supplied along with the plaintext message for encryption, and along with the ciphertext message for decryption. There is a close analogy here with a general-purpose computer and a computer program. The computer, like the cryptosystem, is capable of a large variety of transformations, from which the computer program, like the specific key, selects one. In most cryptosystems, anyone with access to the key can both encrypt and decrypt messages. The key is transmitted to the community of authorized users over a secure channel (as an example, a courier may be used to hand-carry the sensitive key information); the key usually remains unchanged for a considerable number of transmissions. The goal of the *cryptanalyst* (eavesdropper or adversary) is to produce an estimate of the plaintext, $\hat{M}$, by analyzing the ciphertext obtained from the public channel, without benefit of the key.

Encryption schemes fall into two generic categories: *block encryption*, and *data-stream* or simply *stream encryption*. With block encryption, the plaintext is segmented into blocks of fixed size; each block is encrypted independently from the others. For a given key, a particular plaintext block will therefore be carried into the same ciphertext block each time it appears (similar to block encoding). With data-stream encryption, similar to convolutional coding, there is no fixed block size. Each plaintext bit, m_i, is encrypted with the ith element, k_i, of a sequence of symbols (key stream) generated with the key. The encryption is *periodic* if the key stream repeats itself after p characters for some fixed p; otherwise, it is nonperiodic.

In general, the properties desired in an encryption scheme are quite different from those desired in a channel coding scheme. For example, with encryption, plaintext data should never appear directly in the ciphertext, but with channel coding, codes are often in *systematic form* comprised of unaltered message bits

plus parity bits (see Section 5.4.5). Consider another example of the differences between encryption and channel coding. With block encryption, a single bit error at the input of the decryptor might change the value of many of the output bits in the block. This effect, known as *error propagation*, is often a desirable cryptographic property since it makes it difficult for unauthorized users to succeed in spoofing a system. However, in the case of channel coding, we would like the system to correct as many errors as possible, so that the output is relatively unaffected by input errors.

12.1.2 System Goals

The major requirements for a cryptosystem can be stated as follows:

1. To provide an *easy* and *inexpensive* means of encryption and decryption to all authorized users in possession of the appropriate key
2. To ensure that the cryptanalyst's task of producing an estimate of the plaintext without benefit of the key is made *difficult* and *expensive*

Successful cryptosystems are classified as being either *unconditionally secure* or *computationally secure*. A system is said to be *unconditionally secure* when the amount of information available to the cryptanalyst is insufficient to determine the encryption and decryption transformations, no matter how much computing power the cryptanalyst has available. One such system, called a *one-time pad*, involves encrypting a message with a random key that is used one time only. The key is never reused; hence the cryptanalyst is denied information that might be useful against subsequent transmissions with the same key. Although such a system is unconditionally secure (see Section 12.2.1), it has limited use in a conventional communication system, since a new key would have to be distributed for each new message—a great logistical burden. The distribution of keys to the authorized users is a major problem in the operation of any cryptosystem, even when a key is used for an extended period of time. Although some systems can be proven to be unconditionally secure, currently there is no known way to demonstrate security for an arbitrary cryptosystem. Hence the specifications for most cryptosystems rely on the less formal designation of *computational security* for x number of years, which means that under circumstances favorable to the cryptanalyst (i.e., using state-of-the-art computers) the system security could be broken in a period of x years, but could not be broken in less than x years.

12.1.3 Classic Threats

The weakest classification of cryptanalytic threat on a system is called a *ciphertext-only attack*. In this attack the cryptanalyst might have *some* knowledge of the general system and the language used in the message, but the only significant data available to him is the encrypted transmission intercepted from the public channel.

A more serious threat to a system is called a *known plaintext attack*; it involves knowledge of the plaintext *and* knowledge of its ciphertext counterpart. The rigid structure of most business forms and programming languages often provides an opponent with much a priori knowledge of the details of the plaintext message. Armed with such knowledge and with a ciphertext message, the cryptanalyst can mount a known plaintext attack. In the diplomatic arena, if an encrypted message directs a foreign minister to make a particular public statement, and if he does so without paraphrasing the message, the cryptanalyst may be privy to both the ciphertext *and* its exact plaintext translation. While a known plaintext attack is not always possible, its occurrence is frequent enough that a system is not considered secure unless it is designed to be secure against the plaintext attack [2].

When the cryptanalyst is in the position of *selecting* the plaintext, the threat is termed a *chosen plaintext attack*. Such an attack was used by the United States to learn more about the Japanese cryptosystem during World War II. On May 20, 1942, Admiral Yamamoto, Commander-in-Chief of the Imperial Japanese Navy, issued an order spelling out the detailed tactics to be used in the assault of Midway island. This order was intercepted by the Allied listening posts. By this time, the Americans had learned enough of the Japanese code to decrypt most of the message. Still in doubt, however, were some important parts, such as the *place* of the assault. They suspected that the characters "AF" meant Midway island, but to be sure, Joseph Rochefort, head of the Combat Intelligence Unit, decided to use a chosen plaintext attack to trick the Japanese into providing concrete proof. He had the Midway garrison broadcast a distinctive plaintext message in which Midway reported that its fresh-water distillation plant had broken down. The American cryptanalysts needed to wait only two days before they intercepted a Japanese ciphertext message stating that AF was short of fresh water [1].

12.1.4 Classic Ciphers

One of the earliest examples of a monoalphabetic cipher was the *Caesar Cipher*, used by Julius Caesar during the Gallic wars. Each plaintext letter is replaced with a new letter obtained by an *alphabetic shift*. Figure 12.2a illustrates such an encryption transformation, consisting of three end-around shifts of the alphabet. When using this Caesar's alphabet, the message, "now is the time" is encrypted as follows:

Plaintext: N O W I S T H E T I M E

Ciphertext: Q R Z L V W K H W L P H

The decryption key is simply the number of alphabetic shifts; the code is changed by choosing a new key. Another classic cipher system, illustrated in Figure 12.2b, is called the *Polybius square*. Letters I and J are first combined and treated as a single character since the final choice can easily be decided from the context of the message. The resulting 25 character alphabet is arranged in a 5 × 5 array.

Plaintext:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Chiphertext:	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

(a)

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	IJ	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

(b)

Figure 12.2 (a) Caesar's alphabet with a shift of 3. (b) Polybius square.

Encryption of any character is accomplished by choosing the appropriate row–column (or column-row) number pair. An example of encryption with the use of the Polybius square follows:

Plaintext: N O W I S T H E T I M E

Ciphertext: 33 43 25 42 34 44 32 51 44 42 23 51

The code is changed by a rearrangement of the letters in the 5×5 array.

The *Trithemius progressive key*, shown in Figure 12.3, is an example of a *polyalphabetic cipher*. The row labeled shift 0 is identical to the usual arrangement of the alphabet. The letters in the next row are shifted one character to the left with an end-around shift for the leftmost position. Each successive row follows the same pattern of shifting the alphabet one character to the left as compared to the prior row. This continues until the alphabet has been depicted in all possible arrangements of end-around shifts. One method of using such an alphabet is to select the first cipher character from the shift 1 row, the second cipher character from the shift 2 row, and so on. An example of such encryption is

Plaintext: N O W I S T H E T I M E

Ciphertext: O Q Z M X Z O M C S X Q

There are several interesting ways that the Trithemius progressive key can be used. One way, called the *Vigenere key method*, employs a keyword. The key dictates the row choices for encryption and decryption of each successive character in the message. For example, suppose that the word "TYPE" is selected as the key; then an example of the Vigenere encryption method is

Key: T Y P E T Y P E T Y P E

Plaintext: N O W I S T H E T I M E

Ciphertext: G M L M L R W I M G B I

Plaintext:	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Shift:																										
0	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
2	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
3	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
4	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
5	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
6	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
7	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
8	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
9	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
10	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
11	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
12	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
13	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
14	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
15	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
16	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
17	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
18	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
19	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
20	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
21	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
22	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
23	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
24	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
25	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Figure 12.3 Trithemius progressive key.

where the first letter, T, of the key indicates that the row choice for encrypting the first plaintext character is the row starting with T (shift 19). The next row choice starts with Y (shift 24), and so on. A variation of this key method, called the *Vigenere auto (plain) key method*, starts with a single letter or word used as a *priming key*. The priming key dictates the starting row or rows for encrypting the first or first few plaintext characters, as in the preceding example. Next, the *plaintext characters* themseleves are used as the key for choosing the rows for encryption. An example using the letter "F" as the priming key follows:

Key:	F N O W I S T H E T I M
Plaintext:	N O W I S T H E T I M E
Ciphertext:	S B K E A L A L X B U Q

With the auto key method, it should be clear that feedback has been introduced

to the encryption process. With this feedback, the choice of the ciphertext is dictated by the contents of the message.

A final variation of the Vigenere method, called the *Vigenere auto (cipher) key method*, is similar to the plain key method in that a priming key and feedback are used. The difference is that after encryption with the priming key, each successive key character in the sequence is obtained from the prior *ciphertext character* instead of from the plaintext character. An example should make this clear. As before, the letter "F" is used as the priming key.

Key:	F S G C K C V C G Z H T
Plaintext:	N O W I S T H E T I M E
Ciphertext:	S G C K C V C G Z H T X

Although each key character can be found from its preceding ciphertext character, it is functionally dependent on *all* the preceding characters in the message plus the priming key. This has the effect of diffusing the statistical properties of the plaintext across the ciphertext, making statistical analysis very difficult for a cryptanalyst. One weakness of the cipher key example depicted here is that the ciphertext contains key characters which will be exposed on the public channel "for all to see." Variations of this method can be employed to prevent such overt exposure [3]. By today's standards Vigenere's encryption schemes are not very secure; his basic contribution was the discovery that nonrepeating key sequences could be generated by using the messages themselves or functions of the messages.

12.2 THE SECRECY OF A CIPHER SYSTEM

12.2.1 Perfect Secrecy

Consider a cipher system with a finite message space $\{M\} = M_0, M_1, \ldots, M_{N-1}$ and a finite ciphertext space $\{C\} = C_0, C_1, \ldots, C_{U-1}$. For any M_i, the a priori probability that M_i is transmitted is $P(M_i)$. Given that C_j is received, the a posteriori probability that M_i was transmitted is $P(M_i|C_j)$. A cipher system is said to have *perfect secrecy* if for every message M_i and every ciphertext C_j, the a posteriori probability is equal to the a priori probability:

$$P(M_i|C_j) = P(M_i) \tag{12.2}$$

Thus for a system with perfect secrecy, a cryptanalyst who intercepts C_j obtains no further information to enable him or her to determine which message was transmitted. A necessary and sufficient condition for perfect secrecy is that for every M_i and C_j,

$$P(C_j|M_i) = P(C_j) \tag{12.3}$$

The schematic in Figure 12.4 illustrates an example of perfect secrecy. In this example, $\{M\} = M_0, M_1, M_2, M_3$, $\{C\} = C_0, C_1, C_2, C_3$, $\{K\} = K_0, K_1, K_2, K_3$, $N = U = 4$, and $P(M_i) = P(C_j) = \frac{1}{4}$. The transformation from message

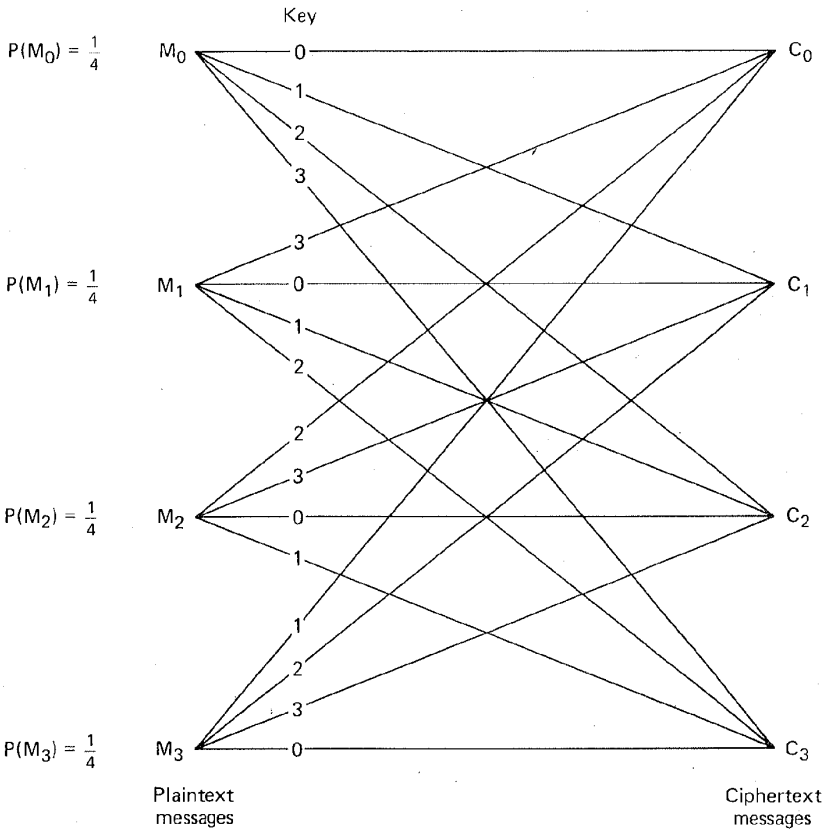


Figure 12.4 Example of perfect secrecy.

to ciphertext is obtained by

$$C_s = T_{K_j}(M_i) \qquad (12.4)$$
$$s = (i + j) \text{ modulo-}N$$

where T_{K_j} indicates a transformation under the key, K_j, and x modulo-y is defined as the remainder of dividing x by y. Thus $s = 0, 1, 2, 3$. A cryptanalyst intercepting one of the ciphertext messages $C_s = C_0$, C_1, C_2, or C_3 would have no way of determining which of the four keys was used, and therefore whether the correct message is M_0, M_1, M_2, or M_3. A cipher system in which the number of messages, the number of keys, and the number of ciphertext transformations are all equal is said to have perfect secrecy if and only if the following two conditions are met:

1. There is only one key transforming each message to each ciphertext.
2. All keys are equally likely.

If these conditions are not met, there would be some message M_i such that

for a given C_j, there is no key that can decipher C_j into M_i, implying that $P(M_i|C_j) = 0$ for some i and j. The cryptanalyst could then eliminate certain plaintext messages from consideration, thereby simplifying the task. Perfect secrecy is a very desirable objective since it means that the cipher system is unconditionally secure. It should be apparent, however, that for systems which transmit a large number of messages, the amount of key that must be distributed for perfect secrecy can result in formidable management problems, making such systems impractical. Since in a system with perfect secrecy, the number of different keys is at least as great as the number of possible messages, if we allow messages of unlimited length, perfect secrecy requires an infinite amount of key.

Example 12.1 Breaking a Cipher System When the Key Space Is Smaller Than the Message Space

Consider that the 29-character ciphertext

G R O B O K B O D R O R O B Y O C Y P I O C D O B I O K B

was produced by a Caesar cipher (see Section 12.1.4) such that each letter has been shifted by K positions, where $1 \leq K \leq 25$. Show how a cryptanalyst can break this code.

Solution

Because the number of possible keys (there are 25) is smaller than the number of possible 29-character meaningful messages (there are a myriad), perfect secrecy cannot be achieved. In the original polyalphabetic cipher of Figure 12.3, a plaintext character is replaced by a letter of increasingly higher rank as the row number (K) increases. Hence, in analyzing the ciphertext, we reverse the process by creating rows such that each ciphertext letter is replaced by letters of decreasing rank. The cipher is easily broken by trying all the keys, from 1 to 25, as shown in Figure 12.5, yielding only one key ($K = 10$) that produces the meaningful message: WHERE ARE THE HEROES OF YESTERYEAR (The spaces have been added.)

Example 12.2 Perfect Secrecy

We can modify the key space of Example 12.1 to create a cipher having perfect secrecy. In this new cipher system each character in the message is encrypted using a *randomly selected* key value. The key, K, is now given by the sequence $k_1, k_2, \ldots, k_{29}$, where each k_i is a random integer in the range (1, 25) dictating the shift used for the ith character; thus there are a total of $(25)^{29}$ different key sequences. Then the 29-character ciphertext in Example 12.1 could correspond to *any* meaningful 29-character message. For example, the ciphertext could correspond to the plaintext (the spaces have been added)

ENGLISH AND FRENCH ARE SPOKEN HERE

derived by the key: 2, 4, 8, 16, 6, 18, 20, ... Most of the 29-character possibilities can be ruled out because they are not meaningful messages (this much is known without the ciphertext). Perfect secrecy is achieved because interception of the ciphertext in this system reveals no additional information about the plaintext message.

Key	Text																											
0	G	R	O	B	O	K	B	O	D	R	O	R	O	B	Y	O	C	Y	P	I	O	C	D	O	B	I	O	K	B
1	F	Q	N	A	N	J	A	N	C	Q	N	Q	N	A	X	N	B	X	O	H	N	B	C	N	A	H	N	J	A
2	E	P	M	Z	M	I	Z	M	B	P	M	P	M	Z	W	M	A	W	N	G	M	A	B	M	Z	G	M	I	Z
3	D	O	L	Y	L	H	Y	L	A	O	L	O	L	Y	V	L	Z	V	M	F	L	Z	A	L	Y	F	L	H	Y
4	C	N	K	X	K	G	X	K	Z	N	K	N	K	X	U	K	Y	U	L	E	K	Y	Z	K	X	E	K	G	X
5	B	M	J	W	J	F	W	J	Y	M	J	M	J	W	T	J	X	T	K	D	J	X	Y	J	W	D	J	F	W
6	A	L	I	V	I	E	V	I	X	L	I	L	I	V	S	I	W	S	J	C	I	W	X	I	V	C	I	E	V
7	Z	K	H	U	H	D	U	H	W	K	H	K	H	U	R	H	V	R	I	B	H	V	W	H	U	B	H	D	U
8	Y	J	G	T	G	C	T	G	V	J	G	J	G	T	Q	G	U	Q	H	A	G	U	V	G	T	A	G	C	T
9	X	I	F	S	F	B	S	F	U	I	F	I	F	S	P	F	T	P	G	Z	F	T	U	F	S	Z	F	B	S
10	W	H	E	R	E	A	R	E	T	H	E	H	E	R	O	E	S	O	F	Y	E	S	T	E	R	Y	E	A	R
11	V	G	D	Q	D	Z	Q	D	S	G	D	G	D	Q	N	D	R	N	E	X	D	R	S	D	Q	X	D	Z	Q
12	U	F	C	P	C	Y	P	C	R	F	C	F	C	P	M	C	Q	M	D	W	C	Q	R	C	P	W	C	Y	P
13	T	E	B	O	B	X	O	B	Q	E	B	E	B	O	L	B	P	L	C	V	B	P	Q	B	O	V	B	X	O
14	S	D	A	N	A	W	N	A	P	D	A	D	A	N	K	A	O	K	B	U	A	O	P	A	N	U	A	W	N
15	R	C	Z	M	Z	V	M	Z	O	C	Z	C	Z	M	J	Z	N	J	A	T	Z	N	O	Z	M	T	Z	V	M
16	Q	B	Y	L	Y	U	L	Y	N	B	Y	B	Y	L	I	Y	M	I	Z	S	Y	M	N	Y	L	S	Y	U	L
17	P	A	X	K	X	T	K	X	M	A	X	A	X	K	H	X	L	H	Y	R	X	L	M	X	K	R	X	T	K
18	O	Z	W	J	W	S	J	W	L	Z	W	Z	W	J	G	W	K	G	X	Q	W	K	L	W	J	Q	W	S	J
19	N	Y	V	I	V	R	I	V	K	Y	V	Y	V	I	F	V	J	F	W	P	V	J	K	V	I	P	V	R	I
20	M	X	U	H	U	Q	H	U	J	X	U	X	U	H	E	U	I	E	V	O	U	I	J	U	H	O	U	Q	H
21	L	W	T	G	T	P	G	T	I	W	T	W	T	G	D	T	H	D	U	N	T	H	I	T	G	N	T	P	G
22	K	V	S	F	S	O	F	S	H	V	S	V	S	F	C	S	G	C	T	M	S	G	H	S	F	M	S	O	F
23	J	U	R	E	R	N	E	R	G	U	R	U	R	E	B	R	F	B	S	L	R	F	G	R	E	L	R	N	E
24	I	T	Q	D	Q	M	D	Q	F	T	Q	T	Q	D	A	Q	E	A	R	K	Q	E	F	Q	D	K	Q	M	D
25	H	S	P	C	P	L	C	P	E	S	P	S	P	C	Z	P	D	Z	Q	J	P	D	E	P	C	J	P	L	C

Figure 12.5 Example of breaking a cipher system when the key space is smaller than the message space.

12.2.2 Entropy and Equivocation

As discussed in Chapter 7, the amount of information in a message is related to the probability of occurrence of the message. Messages with probability of either 0 or 1 contain no information, since we can be very confident concerning our prediction of their occurrence. The more uncertainty there is in predicting the occurrence of a message, the greater is the information content. Hence when each of the messages in a set is equally likely, we can have *no* confidence in our ability to predict the occurrence of a particular message, and the uncertainty or information content of the message is maximum.

Entropy, $H(X)$, is defined as the average amount of information per message. It can be considered a measure of how much *choice* is involved in the selection

of a message, X. It is expressed by the following summation over all possible messages:

$$H(X) = -\sum_X P(X) \log_2 P(X) = \sum_X P(X) \log_2 \frac{1}{P(X)} \tag{12.5}$$

When the logarithm is taken to the base 2, as shown, $H(X)$ is the *expected number of bits* in an *optimally encoded* message, X. This is not quite the measure that a cryptanalyst desires. He will have intercepted some ciphertext and will want to know how confidently he can predict a message (or key) given that this particular ciphertext was sent. *Equivocation*, $H(X|Y)$, defined as the conditional entropy of X given Y, is a more useful measure for the cryptanalyst in attempting to break the cipher.

$$\begin{aligned} H(X|Y) &= -\sum_{X,Y} P(X, Y) \log_2 P(X|Y) \\ &= \sum_Y P(Y) \sum_X P(X|Y) \log_2 \frac{1}{P(X|Y)} \end{aligned} \tag{12.6}$$

Equivocation can be thought of as the uncertainty that message X was sent, having received Y. The cryptanalyst would like $H(X|Y)$ to approach zero as the amount of intercepted ciphertext, Y, increases.

Example 12.3 Entropy and Equivocation

Consider a sample message set consisting of eight equally likely messages $\{X\} = X_1, X_2, \ldots, X_8$.

(a) Find the entropy associated with a message from the set $\{X\}$.
(b) Given another equally likely message set $\{Y\} = Y_1, Y_2$. Consider that the occurrence of each message Y narrows the possible choices of X in the following way:

If Y_1 is present: only X_1, X_2, X_3, or X_4 is possible

If Y_2 is present: only X_5, X_6, X_7, or X_8 is possible

Find the equivocation of message X conditioned on message Y.

Solution

(a) $P(X) = \frac{1}{8}$
$H(X) = 8[(\frac{1}{8}) \log_2 8] = 3$ bits/message

(b) $P(Y) = \frac{1}{2}$. For each Y, $P(X|Y) = \frac{1}{4}$ for four of the X's and $P(X|Y) = 0$ for the remaining four X's. Using Equation (12.6), we obtain

$$H(X|Y) = 2[(\tfrac{1}{2})4(\tfrac{1}{4} \log_2 4)] = 2 \text{ bits/message}$$

We see that knowledge of Y has reduced the uncertainty of X from 3 bits/message to 2 bits/message.

12.2.3 Rate of a Language and Redundancy

The *true rate* of a language, r, is defined as the average number of *information bits* contained in each character and is expressed for messages of length N by

$$r = \frac{H(X)}{N} \tag{12.7}$$

where $H(X)$ is the message entropy, or the number of bits in the *optimally encoded* message. For large N, estimates of r for written English range between 1.0 and 1.5 bits/character [4]. The *absolute rate* or maximum entropy, r', of a language is defined as the maximum number of information bits contained in each character assuming that all possible sequences of characters are equally likely. The absolute rate is given by

$$r' = \log_2 L \tag{12.8}$$

where L is the number of characters in the language. For the English alphabet $r' = \log_2 26 = 4.7$ bits/character. The true rate of English is, or course, much less than its absolute rate since, like most languages, English is highly redundant and structured.

The *redundancy*, D, of a language is defined in terms of its true rate and absolute rate as follows:

$$D = r' - r \tag{12.9}$$

For the English language with $r' = 4.7$ bits/character and $r = 1.5$ bits/character, $D = 3.2$, and the ratio $D/r' = 0.68$ is a measure of the redundancy in the language.

12.2.4 Unicity Distance and Ideal Secrecy

We stated earlier that perfect secrecy requires an infinite amount of key if we allow messages of unlimited length. With a finite key size, the equivocation of the key $H(K|C)$ generally approaches zero, implying that the key can be uniquely determined and the cipher system can be broken. The *unicity distance* is defined as the smallest amount of ciphertext, N, such that the key equivocation $H(K|C)$ is close to zero. Therefore, the unicity distance is the amount of ciphertext needed to uniquely determine the key and thus break the cipher system. Shannon [5] described an *ideal secrecy* system as one in which $H(K|C)$ does not approach zero as the amount of ciphertext approaches infinity; that is, no matter how much ciphertext is intercepted, the key cannot be determined. The term "ideal secrecy" describes a system that does not achieve perfect secrecy but is nonetheless unbreakable (unconditionally secure) because it does not reveal enough information to determine the key.

Most cipher systems are too complex to determine the probabilities required to derive the unicity distance. However, it is sometimes possible to approximate unicity distance, as shown by Shannon [5] and Hellman [6]. Following Hellman, assume that each plaintext and ciphertext message comes from a finite alphabet of L symbols. Thus there are $2^{r'N}$ possible messages of length N, where r' is the

absolute rate of the language. We can consider the total message space partitioned into two classes, meaningful messages, M_1, and meaningless messages M_2:

$$\text{number of meaningful messages} = 2^{rN} \tag{12.10}$$

$$\text{number of meaningless messages} = 2^{r'N} - 2^{rN} \tag{12.11}$$

where r is the true rate of the language, and where the a priori probabilities of the message classes are

$$P(M_1) = \frac{1}{2^{rN}} = 2^{-rN} \qquad M_1 \text{ meaningful} \tag{12.12}$$

$$P(M_2) = 0 \qquad M_2 \text{ meaningless} \tag{12.13}$$

Let us assume that there are $2^{H(K)}$ possible keys (size of the key alphabet), where $H(K)$ is the entropy of the key (number of bits in the key). Assume that all keys are equally likely, that is,

$$P(K) = \frac{1}{2^{H(K)}} = 2^{-H(K)} \tag{12.14}$$

The derivation of the unicity distance is based on a *random cipher* model, which states that for each key K and ciphertext C, the decryption operation $D_K(C)$ yields an independent random variable distributed over all the possible $2^{r'N}$ messages (both meaningful and meaningless). Therefore, for a given K and C, the $D_K(C)$ operation can produce any one of the plaintext messages with equal probability.

Given an encryption described by $C_i = E_{K_i}(M_i)$, a *false solution*, F, arises whenever encryption under another key K_j could also produce C_i either from the message M_i or from some other message M_j; that is,

$$C_i = E_{K_i}(M_i) = E_{K_j}(M_i) = E_{K_j}(M_j) \tag{12.15}$$

A cryptanalyst intercepting C_i would not be able to pick the correct key and hence could not break the cipher system. We are not concerned with the decryption operations that produce *meaningless* messages because these are easily rejected.

For every correct solution to a particular ciphertext there are $2^{H(K)} - 1$ incorrect keys, each of which has the same probability $P(F)$ of yielding a false solution. Because each meaningful plaintext message is assumed equally likely, the probability of a false solution, $P(F)$, is the same as the probability of getting a meaningful message.

$$P(F) = \frac{2^{rN}}{2^{r'N}} = 2^{(r-r')N} = 2^{-DN} \tag{12.16}$$

where $D = r' - r$ is the redundancy of the language. The expected number of false solutions $\overline{F}$ is then

$$\overline{F} = [2^{H(K)} - 1]P(F) = [2^{H(K)} - 1]2^{-DN} \simeq 2^{H(K)-DN} \tag{12.17}$$

Because of the rapid decrease of $\overline{F}$ with increasing N,

$$\log_2 \overline{F} = H(K) - DN = 0 \tag{12.18}$$

is defined as the point where the number of false solutions is sufficiently small so that the cipher can be broken. The resulting unicity distance is therefore

$$N = \frac{H(K)}{D} \tag{12.19}$$

We can see from Equation (12.17) that if $H(K)$ is much larger than DN, there will be a large number of meaningful decryptions, and thus a small likelihood of a cryptanalyst distinguishing which meaningful message is the correct message. In a loose sense, DN represents the number of equations available for solving for the key, and $H(K)$ the number of unknowns. When the number of equations is smaller than the number of unknown key bits, a unique solution is not possible and the system is said to be unbreakable. When the number of equations is larger than the number of unknowns, a unique solution is possible and the system can no longer be characterized as unbreakable (although it may still be computationally secure).

It is the predominance of meaningless decryptions that enables cryptograms to be broken. Equation (12.19) indicates the value of using *data compression* techniques prior to encryption. Data compression removes redundancy, thereby increasing the unicity distance. Perfect data compression would result in $D = 0$ and $N = \infty$ for any key size.

Example 12.4 Unicity Distance

Calculate the unicity distance for a written English encryption system, where the key is given by the sequence $k_1, k_2, \ldots, k_{29}$, where each k_i is a random integer in the range (1, 25) dictating the shift number (Figure 12.3) for the ith character. Assume that each of the possible key sequences is equally likely.

Solution

There are $(25)^{29}$ possible key sequences, each of which is equally likely. Therefore, using Equations (12.5), (12.8), and (12.19) we have:

Key entropy: $H(K) = \log_2 (25)^{29} \simeq 135$ bits

Absolute rate for English: $r' = \log_2 26 = 4.7$ bits/character

Assumed true rate for English: $r = 1.5$ bits/character

Redundancy: $D = r' - r = 3.2$ bits/character

$$N = \frac{H(K)}{D} = \frac{135}{3.2} \cong 43 \text{ characters}$$

In Example 12.2, perfect secrecy was illustrated using the same type of key sequence described here, with a 29-character message. In this example we see that if the available ciphertext is 43 characters long (which implies that some portion of the key sequence must be used twice), a unique solution may be possible. However, there is no indication as to the computational difficulty in finding the solution. Even

though we have estimated the theoretical amount of ciphertext required to break the cipher, it might be computationally infeasible to accomplish this.

12.3 PRACTICAL SECURITY

For ciphertext sequences greater than the unicity distance any system can be solved, in principle, merely by trying each possible key until the unique solution is obtained. This is completely impractical, however, except when the key is extremely small. For example, for a key configured as a permutation of the alphabet, there are $26! \simeq 4 \times 10^{26}$ possibilities (considered small in the cryptographic context). In an exhaustive search, one might expect to reach the right key at about halfway through the search. If we assume that each trial requires a computation time of 1 μs, the total search time exceeds 10^{12} years. Hence techniques other than a brute-force search (e.g., statistical analysis) must be employed if a cryptanalyst is to have any hope of success.

12.3.1 Confusion and Diffusion

A statistical analysis using the frequency of occurrence of individual characters and character combinations can be used to solve many cipher systems. Shannon [5] suggested two encryption concepts for frustrating the statistical endeavors of the cryptanalyst. He termed these encryption transformations confusion and diffusion. *Confusion* involves substitutions that render the final relationship between the key and ciphertext as complex as possible. This makes it difficult to utilize a statistical analysis to narrow the search to a particular subset of the key variable space. Confusion ensures that the majority of the key is needed to decrypt even very short sequences of ciphertext. *Diffusion* involves transformations that smooth out the statistical differences between characters and between character combinations. An example of diffusion with a 26-letter alphabet is to transform a message sequence $M = M_0, M_1, \ldots$ into a new message sequence $Y = Y_0, Y_1, \ldots$ as follows:

$$Y_n = \sum_{i=0}^{s-1} M_{n+i} \qquad \text{modulo-26} \tag{12.20}$$

where each character in the sequence is regarded as an integer modulo-26, s is some chosen integer, and $n = 1, 2, \ldots$. The new message, Y, will have the same redundancy as the original message, M, but the letter frequencies of Y will be more uniform than in M. The effect is that the cryptanalyst needs to intercept a longer sequence of ciphertext before any statistical analysis can be useful.

12.3.2 Substitution

Substitution encryption techniques, such as the Caesar cipher and the Trithemius progressive key cipher, are widely used in puzzles. Such simple substitution ciphers offer little encryption protection. For a substitution technique to fulfill Shan-

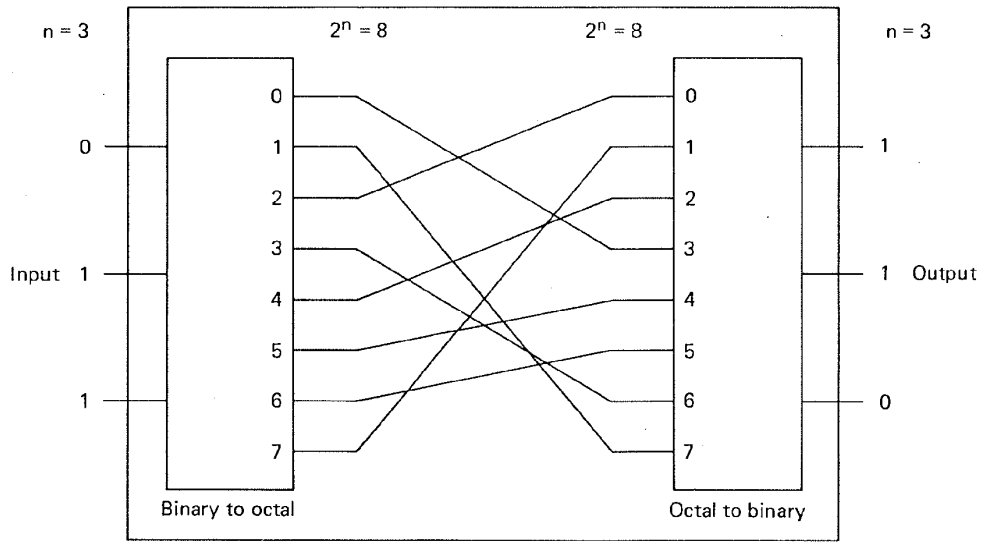
non's concept of *confusion*, a more complex relationship is required. Figure 12.6 shows one example of providing greater substitution complexity through the use of a nonlinear transformation. In general, n input bits are first represented as one of 2^n different characters (binary-to-octal transformation in the example of Figure 12.6). The set of 2^n characters are then permuted so that each character is transposed to one of the others in the set. The character is then converted back to an n-bit output.

It can be easily shown that there are $(2^n)!$ different substitution or connection patterns possible. The cryptanalyst's task becomes computationally unfeasible as n gets large, say $n = 128$; then $2^n = 10^{38}$, and $(2^n)!$ is an astronomical number. We recognize that for $n = 128$, this substitution box (S-box) transformation is complex (confusion). However, although we can identify the S-box with $n = 128$ as ideal, its implementation is not feasible because it would require a unit with $2^n = 10^{38}$ wiring connections.

To verify that the S-box example in Figure 12.6 performs a *nonlinear transformation*, we need only use the superposition theorem stated below as a test. Let

$$\begin{aligned} C &= Ta + Tb \\ C' &= T(a + b) \end{aligned} \tag{12.21}$$

where a and b are input terms, C and C' are output terms, and T is the transformation.



Input	000	001	010	011	100	101	110	111
Output	011	111	000	110	010	100	101	001

Figure 12.6 Substitution box.

If T is linear: $C = C'$ for all inputs

If T is nonlinear: $C \neq C'$

Suppose that $a = 001$ and $b = 010$; then using T as described in Figure 12.6,

$$C = T(001) \oplus T(010) = 111 \oplus 000 = 111$$

$$C' = T(001 \oplus 010) = T(011) = 110$$

where the symbol $\oplus$ represents modulo-2 addition. Since $C \neq C'$, the S-box is nonlinear.

12.3.3 Permutation

In permutation (transposition), the positions of the plaintext letters in the message are simply rearranged, rather than being substituted with other letters of the alphabet as in the classic ciphers. For example, the word THINK might appear, after permutation, as the ciphertext HKTNI. Figure 12.7 represents an example of binary data permutation (a linear operation). Here we see that the input data are simply rearranged or permuted (P-box). The technique has one major disadvantage when used alone; it is vulnerable to trick messages. A trick message is illustrated in Figure 12.7. A single 1 at the input and all the rest 0 quickly reveals one of the internal connections. If the cryptanalyst can subject the system to a

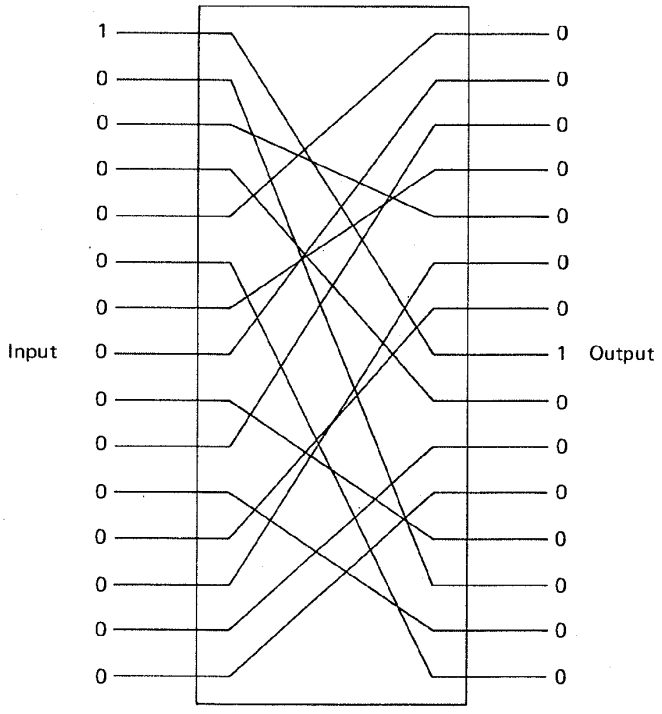


Figure 12.7 Permutation box.

plaintext attack, he will transmit a sequence of such trick messages, moving the single 1 one position for each transmission. In this way, each of the connections from input to output is revealed. This is an example of why a system's security should not depend on its architecture.

12.3.4 Product Cipher System

For transformations involving reasonable numbers of n-message symbols, both of the foregoing cipher systems (the S-box and the P-box) are by themselves wanting. Shannon [5] suggested using a *product cipher* or a combination of S-box and P-box transformations, which together could yield a cipher system more powerful than either one alone. This approach of alternately applying substitution and permutation transformations has been used by IBM in the LUCIFER system [7, 8], and has become the basis for the national Data Encryption Standard (DES) [9]. Figure 12.8 illustrates such a combination of P-boxes and S-boxes. Decryption is accomplished by running the data backward, using the inverse of each S-box. The system as pictured in Figure 12.8 is difficult to implement since each S-box is different, a randomly generated key is not usable, and the system does not lend itself to repeated use of the same circuitry. To avoid these difficulties, the LUCIFER system [8] used two different types of S-boxes, S_1 and S_0, which could be publicly revealed. Figure 12.9 illustrates such a system. The input data are transformed by the sequence of S-boxes and P-boxes under the dictates of a key. The 25-bit key in this example designates, with a binary one or zero, the choice (S_1 or S_0) of each of the 25 S-boxes in the block. The details of the encryption devices can be revealed since security of the system is provided by the key.

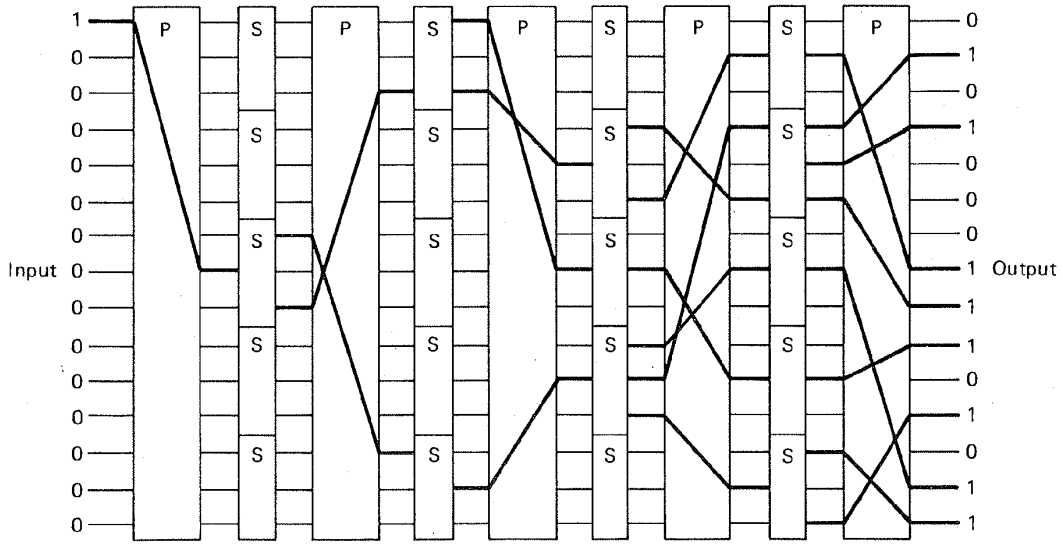


Figure 12.8 Product cipher system.

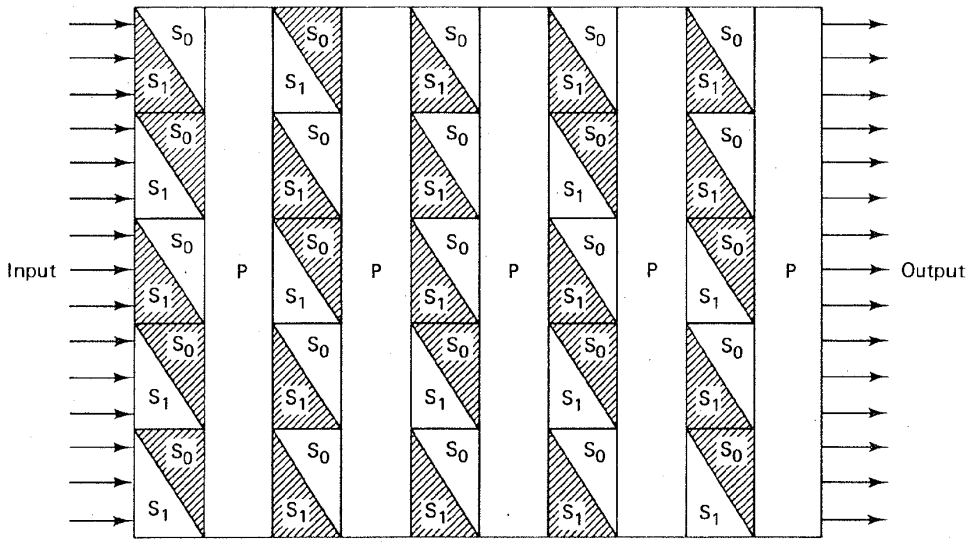


Figure 12.9 Individual keying capability.

The iterated structure of the product cipher system in Figure 12.9 is typical of most present-day block ciphers. The messages are partitioned into successive blocks of n bits, each of which is encrypted with the same key. The n-bit block represents one of 2^n different characters, allowing for $(2^n)!$ different substitution patterns. Consequently, for a reasonable implementation, the substitution part of the encryption scheme is performed in parallel on small segments of the block. An example of this is seen in the next section.

12.3.5 The Data Encryption Standard

In 1977, the National Bureau of Standards adopted a modified Lucifer system as the national Data Encryption Standard (DES) [9]. From a system input–output point of view, DES can be regarded as a block encryption system with an alphabet size of 2^{64} symbols, as shown in Figure 12.10. An input block of 64 bits, regarded

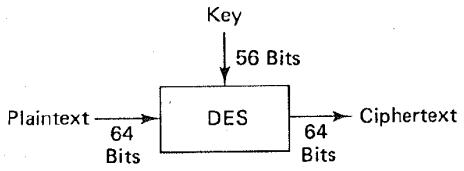


Figure 12.10 Data encryption standard (DES) viewed as a block encryption system.

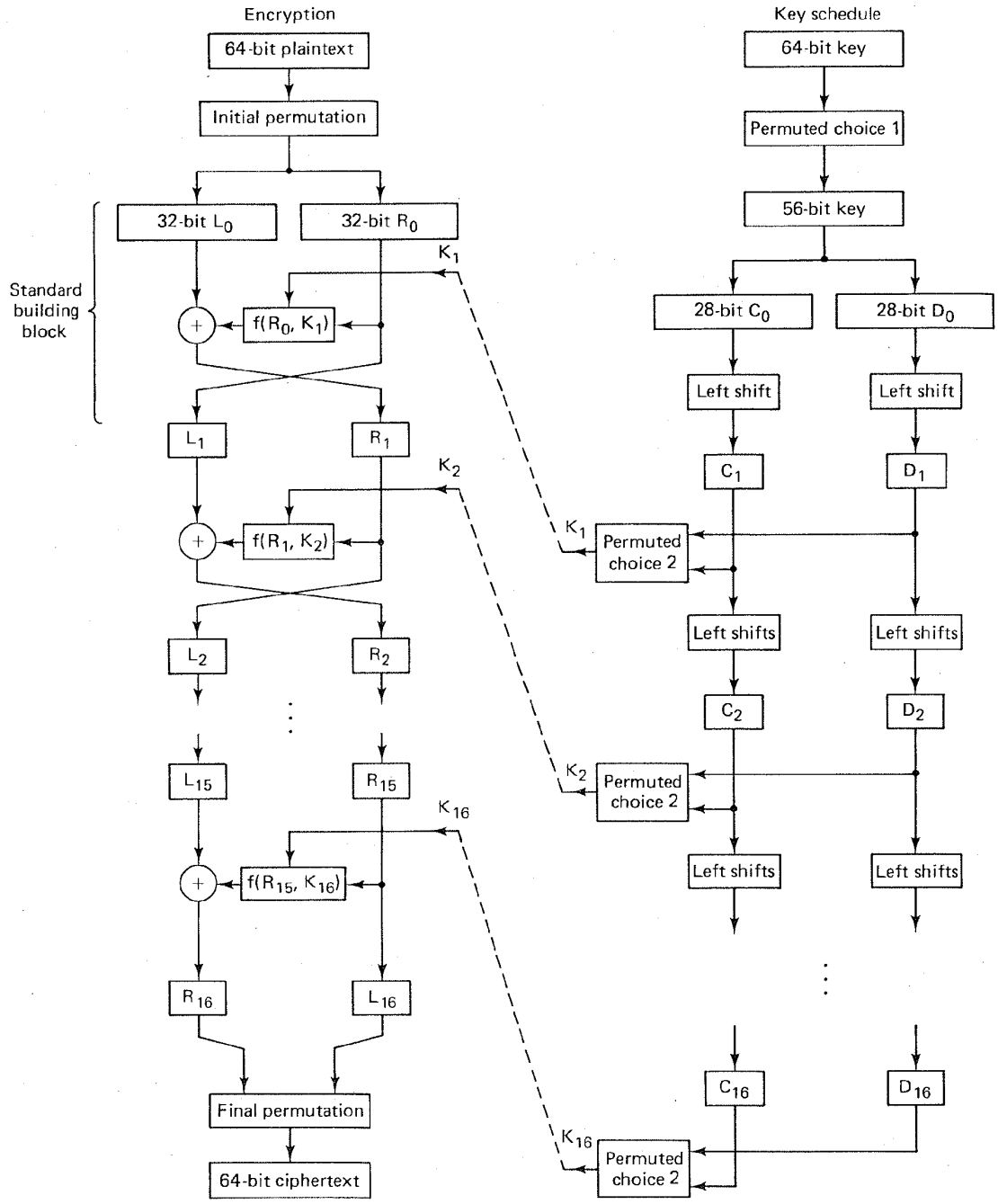


Figure 12.11 Data encryption standard.

as a plaintext symbol in this alphabet, is replaced with a new ciphertext symbol. Figure 12.11 illustrates the system functions in block diagram form. The encryption algorithm starts with an initial permutation (IP) of the 64 plaintext bits, described in the IP-table (Table 12.1). The IP-table is read from left to right and from top to bottom, so that bits $x_1, x_2, \ldots, x_{64}$ are permuted to $x_{58}, x_{50}, \ldots, x_7$. After this initial permutation, the heart of the encryption algorithm consists of 16 iterations using the standard building block (SBB) shown in Figure 12.12. The standard building block uses 48 bits of key to transform the 64 input data bits into 64 output data bits, designated as 32 left-half bits and 32 right-half bits. The output of each building block becomes the input to the next building block. The input right-half 32 bits (R_{i-1}) are copied unchanged to become the output left-half 32 bits (L_i). The R_{i-1} bits are also *extended* and transformed into 48 bits with the E-table (Table 12.2), and then modulo-2 summed with the 48 bits of the key. As in the case of the IP-table, the E-table is read from left to right and from top to bottom. The table expands bits

$$R_{i-1} = x_1, x_2, \ldots, x_{32}$$

into

$$(R_{i-1})_E = x_{32}, x_1, x_2, \ldots, x_{32}, x_1 \tag{12.22}$$

Notice that the bits listed in the first and last columns of the E-table are those bit positions that are used twice to provide the 32 bit-to-48 bit expansion.

Next, $(R_{i-1})_E$ is modulo-2 summed with the ith key selection, explained later, and the result is segmented into eight 6-bit blocks

$$B_1, B_2, \ldots, B_8$$

that is,

$$(R_{i-1})_E \oplus K_i = B_1, B_2, \ldots, B_8 \tag{12.23}$$

Each of the eight 6-bit blocks, B_j, is then used as an input to an S-box function which returns a 4-bit block, $S_j(B_j)$. Thus the input 48 bits are transformed by the S-box to 32 bits. The S-box mapping function, S_j, is defined in Table 12.3. The transformation of $B_j = b_1, b_2, b_3, b_4, b_5, b_6$ is accomplished as follows. The integer corresponding to bits $b_1\ b_6$ selects a row in the table, and the integer corresponding to bits $b_2\ b_3\ b_4\ b_5$ selects a column in the table. For example, if

TABLE 12.1 Initial Permutation (IP)

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

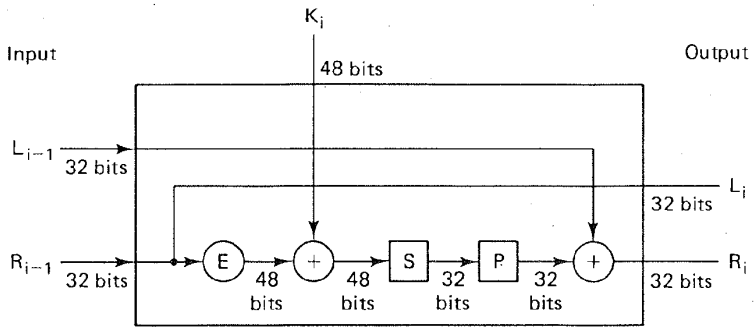


Figure 12.12 Standard building block (SBB).

$b_1 = 110001$, then S_1 returns the value in row 3, column 8, which is the integer 5 and is represented by the bit sequence 0101. The resulting 32-bit block out of the S-box is then permuted using the P-table (Table 12.4). As in the case of the other tables, the P-table is read from left to right and from top to bottom, so that bits $x_1, x_2, \ldots, x_{32}$ are permuted to $x_{16}, x_7, \ldots, x_{25}$. The 32-bit output of the P-table is modulo-2 summed with the input left-half 32 bits (L_{i-1}), forming the output right-half 32 bits (R_i).

The algorithm of the standard building block can be represented by

$$L_i = R_{i-1} \tag{12.24}$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i) \tag{12.25}$$

where $f(R_{i-1}, K_i)$ denotes the functional relationship comprised of the E-table, S-box, and P-table described above. After 16 iterations of the SBB, the data are transposed according to the final inverse permutation (IP^{-1}) described in the IP^{-1}-table (Table 12.5), where the output bits are read from left to right and from top to bottom, as before.

To decrypt, the same algorithm is used but the key sequence that is used in the standard building block is taken in the reverse order. Note that the value of $f(R_{i-1}, K_i)$ which can also be expressed in terms of the output of the ith block as $f(L_i, K_i)$, makes the decryption process possible.

TABLE 12.2 E-Table Bit Selection

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

TABLE 12.3 S-Box Selection Functions

	Column																
Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7	
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8	S_1
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0	
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13	
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10	
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5	S_2
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15	
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9	
0	10	0	9	14	6	3	15	5	1	13	12	17	11	4	2	8	
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1	S_3
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12	
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15	
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9	S_4
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4	
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14	
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6	S_5
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14	
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3	
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11	
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8	S_6
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6	
3	4	3	2	12	9	5	15	0	11	14	1	7	6	0	8	13	
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1	
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6	S_7
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2	
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12	
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7	
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2	S_8
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8	
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11	

TABLE 12.4 P-Table Permutation

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

12.3.5.1 Key Selection

Key selection also proceeds in 16 iterations, as seen in the key schedule portion of Figure 12.11. The input key consists of a 64-bit block with 8 parity bits in positions 8, 16, . . . , 64. The permuted choice 1 (PC-1) discards the parity bits and permutes the remaining 56 bits as shown in Table 12.6. The output of PC-1 is split into two halves, C and D, of 28 bits each. Key selection proceeds in 16 iterations in order to provide a different set of 48 key bits to each SBB encryption iteration. The C and D blocks are successively shifted as follows:

$$C_i = \mathrm{LS}_i(C_{i-1}) \quad \text{and} \quad D_i = \mathrm{LS}_i(D_{i-1}) \tag{12.26}$$

where LS_i is a left circular shift by the number of positions shown in Table 12.7. The sequence C_i, D_i is then transposed according to the permuted choice 2 (PC-2) shown in Table 12.8. The result is the key sequence, K_i, which is used in the ith iteration of the encryption algorithm.

The DES can be implemented as a block encryption system (sce Figure 12.11), which is sometimes referred to as a *codebook* method. A major disadvantage of this method is that a given block of input plaintext will always result in the same output ciphertext (under the same key). Another encryption mode, called the *cipher feedback* mode, encrypts single bits rather than characters, resulting in a stream encryption system [3]. With the cipher feedback scheme (described later), the encryption of a segment of plaintext not only depends on the key and the current data, but also on some of the earlier data.

Since the late 1970s, two points of contention have been widely publicized about the DES [10]. The first concerns the key variable length. Some researchers

TABLE 12.5 Final Permutation (IP^{-1})

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

TABLE 12.6 Key Permutation PC-1

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

TABLE 12.7 Key Schedule of Left Shifts

Iteration, i	Number of left shifts
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

TABLE 12.8 Key Permutation PC-2

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

felt that 56 bits are not adequate to preclude an exhaustive search. The second concerns the details of the internal structure of the S-boxes, which were never released by IBM. The National Security Agency (NSA), which had been involved in the testing of the DES algorithm, had requested that the information not be publicly discussed, since it was sensitive. The critics feared that NSA had been involved in design selections that would allow NSA to "tap into" any DES-encrypted messages [10].

12.4 STREAM ENCRYPTION

Earlier, we defined a *one-time pad* as an encryption system with a random key, used one time only, that exhibits unconditional security. One can conceptualize a stream encryption implementation of a one-time pad using a truly random key stream (the key sequence never repeats). Thus perfect secrecy can be achieved for an infinite number of messages, since each message would be encrypted with a different portion of the random key stream. The development of stream encryption schemes represents an attempt to emulate the one-time pad. Great emphasis was placed on generating key streams that appeared to be random, yet could easily be implemented for decryption, because they could be generated by algorithms. Such stream encryption techniques use pseudorandom (PN) sequences, which derive their name from the fact that they appear random to the casual observer; binary pseudorandom sequences have statistical properties similar to the random flipping of a fair coin. However, the sequences, of course, are deterministic (see Section 10.2). These techniques are popular because the encryption and decryption algorithms are readily implemented with feedback shift registers. At first glance it may appear that a PN key stream can provide the same security as the one-time pad, since the period of the sequence generated by a maximum-length linear shift register is $2^n - 1$ bits, where n is the number of stages in the register. If the PN sequence were implemented with a 50-stage register and a 1-MHz clock rate, the sequence would repeat every $2^{50} - 1$ microseconds, or every 35 years. In this era of large-scale integrated (LSI) circuits, it is just as easy to provide an implementation with 100 stages, in which case the sequence would repeat every 4×10^{16} years. Therefore, one might suppose that since the PN sequence does not repeat itself for such a long time, it would appear truly random and yield perfect secrecy. There is one important difference between the PN sequence and a truly random sequence used by a one-time pad. The PN sequence is generated by an algorithm; thus, knowing the algorithm, one knows the entire sequence. In Section 12.4.2 we will see that an encryption scheme that uses a linear feedback shift register in this way is very vulnerable to a *known plaintext attack.*

12.4.1 Example of Key Generation Using a Linear Feedback Shift Register

Stream encryption techniques generally employ shift registers for generating their PN key sequences. A shift register can be converted into a pseudorandom sequence generator by including a feedback loop that computes a new term for the

first stage based on the previous n terms. The register is said to be linear if the numerical operation in the feedback path is linear. The PN generator example from Section 10.2 is repeated in Figure 12.13. For this example, it is convenient to number the stages as shown in Figure 12.13, where $n = 4$ and the outputs from stages 1 and 2 are modulo-2 added (linear operation) and fed back to stage 4. If the initial state of stages (x_4, x_3, x_2, x_1) is 1 0 0 0, the succession of states triggered by clock pulses would be 1 0 0 0, 0 1 0 0, 0 0 1 0, 1 0 0 1, 1 1 0 0, and so on. The output sequence is made up of the bits shifted out from the rightmost stage of the register, that is, 1 1 1 1 0 1 0 1 1 0 0 1 0 0 0, where the rightmost bit in this sequence is the earliest output and the leftmost bit is the most recent output. Given any linear feedback shift register of degree n, the output sequence is ultimately periodic.

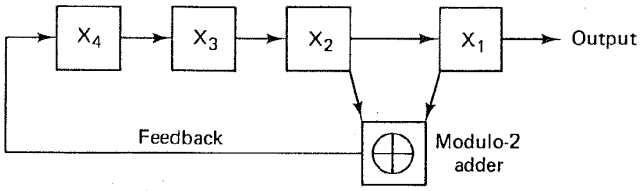


Figure 12.13 Linear feedback shift register example.

12.4.2 Vulnerabilities of Linear Feedback Shift Registers

An encryption scheme that uses a linear feedback shift register (LFSR) to generate the key stream is very vulnerable to attack. A cryptanalyst needs only $2n$ bits of plaintext and its corresponding ciphertext to determine the feedback taps, the initial state of the register, and the entire sequence of the code. In general, $2n$ is very small compared to the period $2^n - 1$. Let us illustrate this vulnerability with the LFSR example illustrated in Figure 12.13. Imagine that a cryptanalyst, who knows nothing about the internal connections of the LFSR, manages to obtain $2n = 8$ bits of ciphertext and its plaintext equivalent. These are shown below, where the rightmost bit is the earliest received and the leftmost bit is the most recent that was received.

Plaintext: 0 1 0 1 0 1 0 1

Ciphertext: 0 0 0 0 1 1 0 0

The cryptanalyst adds the two sequences together, modulo-2, to obtain the segment of the key stream, 0 1 0 1 1 0 0 1, illustrated in Figure 12.14. The key stream sequence shows the contents of the LFSR stages at various times. The rightmost border surrounding four of the key bits shows the contents of the shift register at time t_1. As we successively slide the "moving" border one digit to the left, we see the shift register contents at times $t_2, t_3, t_4, \ldots$. From the linear structure of the four-stage shift register, we can write

$$g_4x_4 + g_3x_3 + g_2x_2 + g_1x_1 = x_5 \tag{12.27}$$

where x_5 is the digit fed back to the input and g_i (= 1 or 0) defines the ith feedback

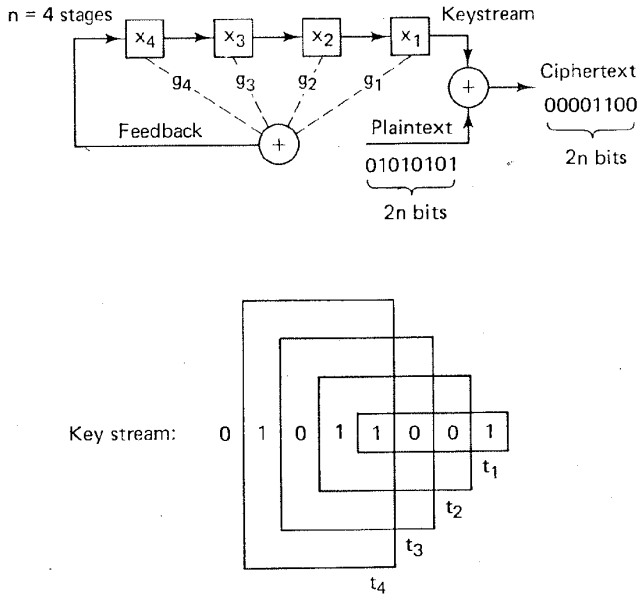


Figure 12.14 Example of vulnerability of a linear feedback shift register.

connection. For this example we can thus write the following four equations with four unknowns, by examining the contents of the shift register at the four times shown in Figure 12.14.

$$\begin{aligned} g_4(1) + g_3(0) + g_2(0) + g_1(1) &= 1 \\ g_4(1) + g_3(1) + g_2(0) + g_1(0) &= 0 \\ g_4(0) + g_3(1) + g_2(1) + g_1(0) &= 1 \\ g_4(1) + g_3(0) + g_2(1) + g_1(1) &= 0 \end{aligned} \tag{12.28}$$

whose solution is $g_1 = 1$, $g_2 = 1$, $g_3 = 0$, $g_4 = 0$, corresponding to the LFSR shown in Figure 12.13. The cryptanalyst has thus learned the connections of the LFSR, together with the starting state of the register at time t_1. He can therefore know the sequence for all time [3]. To generalize this example for any n-stage LFSR, we rewrite Equation (12.27) as follows:

$$x_{n+1} = \sum_{i=1}^{n} g_i x_i \tag{12.29}$$

We can write Equation (12.29) as the matrix equation

$$\mathbf{x} = \mathbf{X}\mathbf{g} \tag{12.30}$$

where

$$\mathbf{x} = \begin{bmatrix} x_{n+1} \\ x_{n+2} \\ \vdots \\ x_{2n} \end{bmatrix} \qquad \mathbf{g} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix}$$

and

$$\mathbf{X} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ x_2 & x_3 & \cdots & x_{n+1} \\ \vdots & \vdots & & \vdots \\ x_n & x_{n+1} & \cdots & x_{2n-1} \end{bmatrix}$$

It can be shown [3] that the columns of $\mathbf{X}$ are linearly independent; thus $\mathbf{X}$ is nonsingular (its determinant is nonzero) and has an inverse. Hence,

$$\mathbf{g} = \mathbf{X}^{-1}\mathbf{x} \tag{12.31}$$

The matrix inversion requires at most on the order of n^3 operations and is thus easily accomplished by computer for any reasonable value of n. For example, if $n = 100$, $n^3 = 10^6$, and a computer with a 1-μs operation cycle would require 1 s for the inversion. The weakness of a LFSR is caused by the linearity of Equation (12.31). The use of *nonlinear feedback* in the shift register makes the cryptanalyst's task much more difficult, if not computationally intractable.

12.4.3 Synchronous and Self-Synchronous Stream Encryption Systems

We can categorize stream encryption systems as either *synchronous* or *self-synchronous*. In the former, the key stream is generated independently of the message, so that a lost character during transmission necessitates a resynchronization of the transmission and receiver key generators. A synchronous stream cipher is shown in Figure 12.15. The starting state of the key generator is initialized with a known input, I_0. The ciphertext is obtained by the modulo addition of the ith key character, k_i, with the ith message character, m_i. Such synchronous ciphers are generally designed to utilize *confusion* (see Section 12.3.1) but not *diffusion*. That is, the encryption of a character is not diffused over some block length of message. For this reason, synchronous stream ciphers do not exhibit *error propagation*.

In a *self-synchronous* stream cipher, each key character is derived from a fixed number, n, of the preceding ciphertext characters, giving rise to the name *cipher feedback*. In such a system, if a ciphertext character is lost during trans-

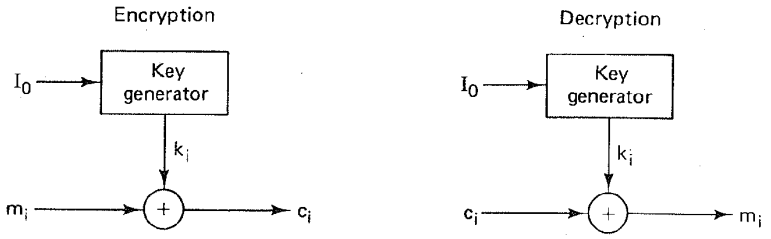


Figure 12.15 Synchronous stream cipher.

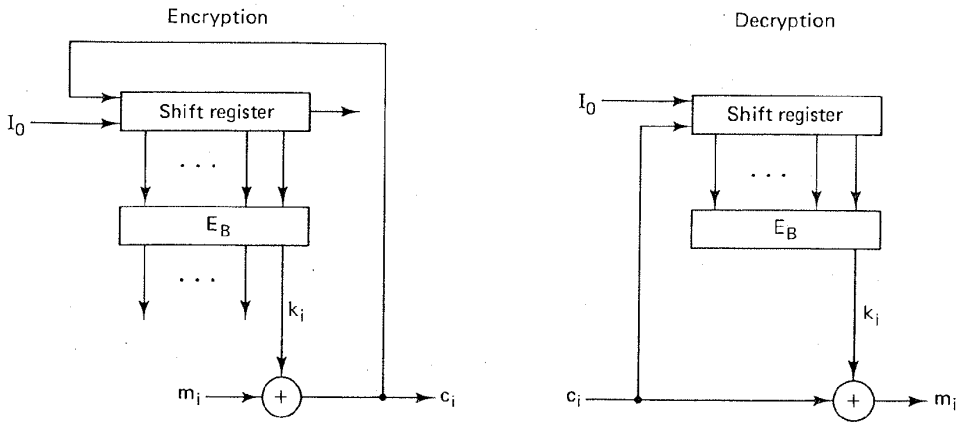


Figure 12.16 Cipher feedback mode.

mission, the error propagates forward for n characters, but the system resynchronizes itself after n correct ciphertext characters are received.

In Section 12.1.4 we looked at an example of cipher feedback in the Vigenere auto key cipher. We saw that the advantages of such a system are that (1) a nonrepeating key is generated, and (2) the statistics of the plaintext message are diffused throughout the ciphertext. However, the fact that the key was exposed in the ciphertext was a basic weakness. This problem can be eliminated by passing the ciphertext characters through a nonlinear block cipher to obtain the key characters. Figure 12.16 illustrates a shift register key generator operating in the cipher feedback mode. Each output ciphertext character, c_i (formed by the modulo addition of the message character, m_i, and the key character, k_i), is fed back to the input of the shift register. As before, initialization is provided by a known input, I_0. At each iteration, the output of the shift register is used as input to a (nonlinear) block encryption algorithm, E_B. The low-order output character from E_B becomes the next key character, k_{i+1}, to be used with the next message character, m_{i+1}. Since, after the first few iterations, the input to the algorithm depends only on the ciphertext, the system is self-synchronizing.

12.5 PUBLIC KEY CRYPTOSYSTEMS

The concept of public key cryptosystems was introduced in 1976 by Diffie and Hellman [11]. In conventional cryptosystems the encryption algorithm can be revealed since the security of the system depends on a safeguarded key. The same key is used for both encryption and decryption. Public key cryptosystems utilize *two different* keys, one for encryption and the other for decryption. In public key cryptosystems, not only the encryption algorithm but also the encryption key can be publicly revealed without compromising the security of the system. In fact, a public directory, much like a telephone directory, is envisioned, which contains the encryption keys of all the subscribers. Only the decryption keys are kept

secret. Figure 12.17 illustrates such a system. The important features of a public key cryptosystem are as follows:

1. The encryption algorithm, E_K, and the decryption algorithm, D_K, are invertible transformations on the plaintext, M, or the ciphertext, C, defined by the key K. That is, for each K and M, if $C = E_K(M)$, then $M = D_K(C) = D_K[E_K(M)]$.
2. For each K, E_K and D_K are easy to compute.
3. For each K, the computation of D_K from E_K is computationally intractable.

Such a system would enable secure communication between subscribers who have never met or communicated before. For example, as seen in Figure 12.17, subscriber A can send a message, M, to subscriber B by looking up B's encryption key in the directory and applying the encryption algorithm, E_B, to obtain the ciphertext $C = E_B(M)$, which he transmits on the public channel. Subscriber B is the only party who can decrypt C by applying his decryption algorithm, D_B, to obtain $M = D_B(C)$.

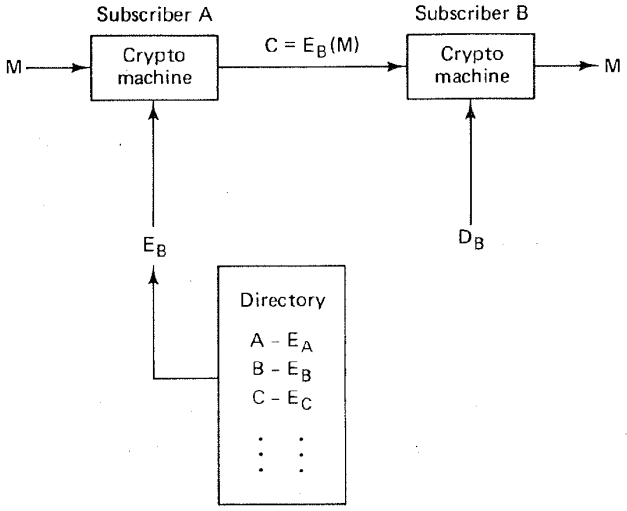


Figure 12.17 Public key cryptosystem.

12.5.1 Signature Authentication Using a Public Key Cryptosystem

Figure 12.18 illustrates the use of a public key cryptosystem for signature authentication. Subscriber A "signs" his message by first applying his decryption algorithm, D_A, to the message, yielding $S = D_A(M) = E_A^{-1}(M)$. Next, he uses the encryption algorithm, E_B, of subscriber B to encrypt S, yielding $C = E_B(S) = E_B[E_A^{-1}(M)]$, which he transmits on a public channel. When subscriber B receives C, he first decrypts it using his private decryption algorithm, D_B, yielding $D_B(C) = E_A^{-1}(M)$. Then he applies the encryption algorithm of subscriber A to produce $E_A[E_A^{-1}(M)] = M$.

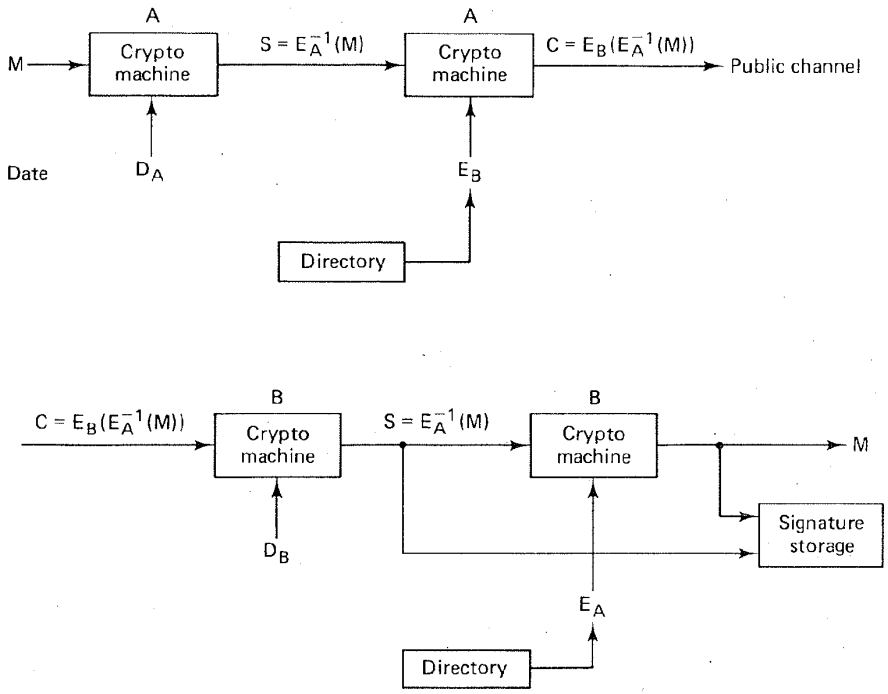


Figure 12.18 Signature authentication using a public key cryptosystem.

If the result is an intelligible message, it must have been initiated by subscriber A, since no one else could have known A's secret decryption key to form $S = D_A(M)$. Notice that S is both message dependent and signer dependent, which means that while B can be sure that the received message indeed came from A, at the same time A can be sure that no one can attribute any false messages to him.

12.5.2 A Trapdoor One-Way Function

Public key cryptosystems are based on the concept of trapdoor one-way functions. Let us first define a *one-way function* as an easily computed function whose inverse is computationally infeasible to find. For example, consider the function $y = x^5 + 12x^3 + 107x + 123$. It should be apparent that given x, y is easy to compute, but given y, x is relatively difficult to compute. A *trapdoor one-way function* is a one-way function whose inverse is easily computed if certain features, used to design the function, are known. Like a trapdoor, such functions are easy to go through in one direction. Without special information the reverse process takes an impossibly long time. We will apply the concept of a trapdoor in Section 12.5.5, when we discuss the Merkle–Hellman scheme.

12.5.3 The Rivest–Shamir–Adelman Scheme

In the Rivest–Shamir–Adelman (RSA) scheme messages are first represented as integers in the range $(0, n-1)$. Each user chooses his own value of n and another pair of positive integers, e and d, in a manner to be described below. The user places his encryption key, the number pair (n, e), in the public directory. The decryption key consists of the number pair (n, d), of which d is kept secret. Encryption of a message, M, and decryption of a ciphertext, C, are defined as follows:

$$\begin{aligned} &\text{Encryption:} \quad C = E(M) = (M)^e \text{ modulo-}n \\ &\text{Decryption:} \quad M = D(C) = (C)^d \text{ modulo-}n \end{aligned} \tag{12.32}$$

They are each easy to compute and the results of each operation are integers in the range $(0, n-1)$. In the RSA scheme, n is obtained by selecting *two large prime numbers*, p and q, and multiplying them together:

$$n = pq \tag{12.33}$$

Although n is made public, p and q are kept hidden, due to the great difficulty in factoring n. Then

$$\phi(n) = (p-1)(q-1) \tag{12.34}$$

called *Euler's totient function*, is formed. The parameter $\phi(n)$ has the interesting property [12] that for any integer X in the range $(0, n-1)$ and any integer k,

$$X = X^{k\phi(n)+1} \text{ modulo-}n \tag{12.35}$$

Therefore, while all other arithmetic is done modulo-n, arithmetic in the exponent is done modulo-$\phi(n)$. A large integer, d, is randomly chosen so that it is relatively prime to $\phi(n)$, which means that $\phi(n)$ and d must have no common divisors other than 1, expressed as

$$\gcd[\phi(n), d] = 1 \tag{12.36}$$

where gcd means "greatest common divisor." Any prime number greater than the larger of (p, q) will suffice. Then the integer e, where $0 < e < \phi(n)$, is found from the following relationship:

$$ed \text{ modulo-}\phi(n) = 1 \tag{12.37}$$

which, from Equation (12.35), is tantamount to choosing e and d to satisfy

$$X = X^{ed} \text{ modulo-}n \tag{12.38}$$

Therefore,

$$E[D(X)] = D[E(X)] = X \tag{12.39}$$

and decryption works correctly. Given an encryption key (n, e), one way that a cryptanalyst might attempt to break the cipher is to factor n into p and q, compute

$\phi(n) = (p - 1)(q - 1)$, and compute d from Equation (12.37). This is all straightforward except for the factoring of n.

The RSA scheme is based on the fact that it is easy to generate two large prime numbers, p and q, and multiply them together, but it is very much more difficult to factor the result. The product can therefore be made public as part of the encryption key, without compromising the factors that would reveal the decryption key corresponding to the encryption key. By making each of the factors roughly 100 digits long, the multiplication can be done in a fraction of a second, but the exhaustive factoring of the result should take billions of years [2].

12.5.3.1 Use of the RSA Scheme

Using the example in Reference [12], let $p = 47$, $q = 59$. Therefore, $n = pq = 2773$ and $\phi(n) = (p - 1)(q - 1) = 2668$. The parameter d is chosen to be relatively prime to $\phi(n)$. For example, choose $d = 157$. Next, the value of e is computed as follows (the details are shown in the next section):

$$ed \text{ modulo } \phi(n) = 1$$

$$157e \text{ modulo } 2668 = 1$$

Therefore, $e = 17$. Consider the plaintext example

ITS ALL GREEK TO ME

By replacing each letter with a two-digit number in the range (01, 26) corresponding to its position in the alphabet, and encoding a blank as 00, the plaintext message can be written as

0920 1900 0112 1200 0718 0505 1100 2015 0013 0500

Each message needs to be expressed as an integer in the range $(0, n - 1)$; therefore, for this example, encryption can be performed on blocks of four digits at a time since this is the maximum number of digits that will always yield a number less than $n - 1 = 2772$. The first four digits (0920) of the plaintext are encrypted as follows:

$$C = (M)^e \text{ modulo-}n = (920)^{17} \text{ modulo-}2773 = 948$$

Continuing this process for the remaining plaintext digits, we get

$$C = 0948\ 2342\ 1084\ 1444\ 2663\ 2390\ 0778\ 0774\ 0219\ 1655$$

The plaintext is returned by applying the decryption key, as follows:

$$M = (C)^{157} \text{ modulo-}2773$$

12.5.3.2 How to Compute e

A variation of Euclid's algorithm [13] for computing the gcd of $\phi(n)$ and d is used to compute e. First, compute a series $x_0, x_1, x_2, \ldots$, where $x_0 = \phi(n)$, $x_1 = d$, and $x_{i+1} = x_{i-1}$ modulo-x_i, until an $x_k = 0$ is found. Then the gcd $(x_0, x_1) = x_{k-1}$. For each x_i compute numbers a_i and b_i such that $x_i = a_i x_0 + b_i x_1$.

If $x_{k-1} = 1$, then b_{k-1} is the multiplicative inverse of x_1 modulo-x_0. If b_{k-1} is a negative number, the solution is $b_{k-1} + \phi(n)$.

Example 12.5 Computation of e from d and $\phi(n)$

For the previous example, with $p = 47$, $q = 59$, $n = 2773$, $\phi(n) = 2688$, and d chosen to be 157, use the Euclid algorithm to verify that $e = 17$.

Solution

i	x_i	a_i	b_i	y_i
0	2668	1	0	
1	157	0	1	16
2	156	1	−16	1
3	1	−1	17	

where

$$y_i = \left\lfloor \frac{x_{i-1}}{x_i} \right\rfloor$$

$$x_{i+1} = x_{i-1} - y_i x_i$$

$$a_{i+1} = a_{i-1} - y_i a_i$$

$$b_{i+1} = b_{i-1} - y_i b_i$$

Hence

$$e = b_3 = 17$$

12.5.4 The Knapsack Problem

The classic knapsack problem is illustrated in Figure 12.19. The knapsack is filled with a subset of the items shown with weights indicated in grams. Given the weight of the filled knapsack (the scale is calibrated to deduct the weight of the empty knapsack), determine which items are contained in the knapsack. For this simple example, the solution can easily be found by trial and error. However, if there are 100 possible items in the set, instead of 10, the problem may become computationally infeasible.

Let us express the knapsack problem in terms of a knapsack vector and a data vector. The knapsack vector is an n-tuple of distinct integers (analogous to the set of possible knapsack items)

$$\mathbf{a} = a_1, a_2, \ldots, a_n$$

The data vector is an n-tuple of binary symbols

$$\mathbf{x} = x_1, x_2, \ldots, x_n$$

The knapsack, S, is the sum of a subset of the components of the knapsack vector

$$S = \sum_{i=1}^{n} a_i x_i \qquad \text{where } x_i = 0, 1$$
$$= \mathbf{ax} \tag{12.40}$$

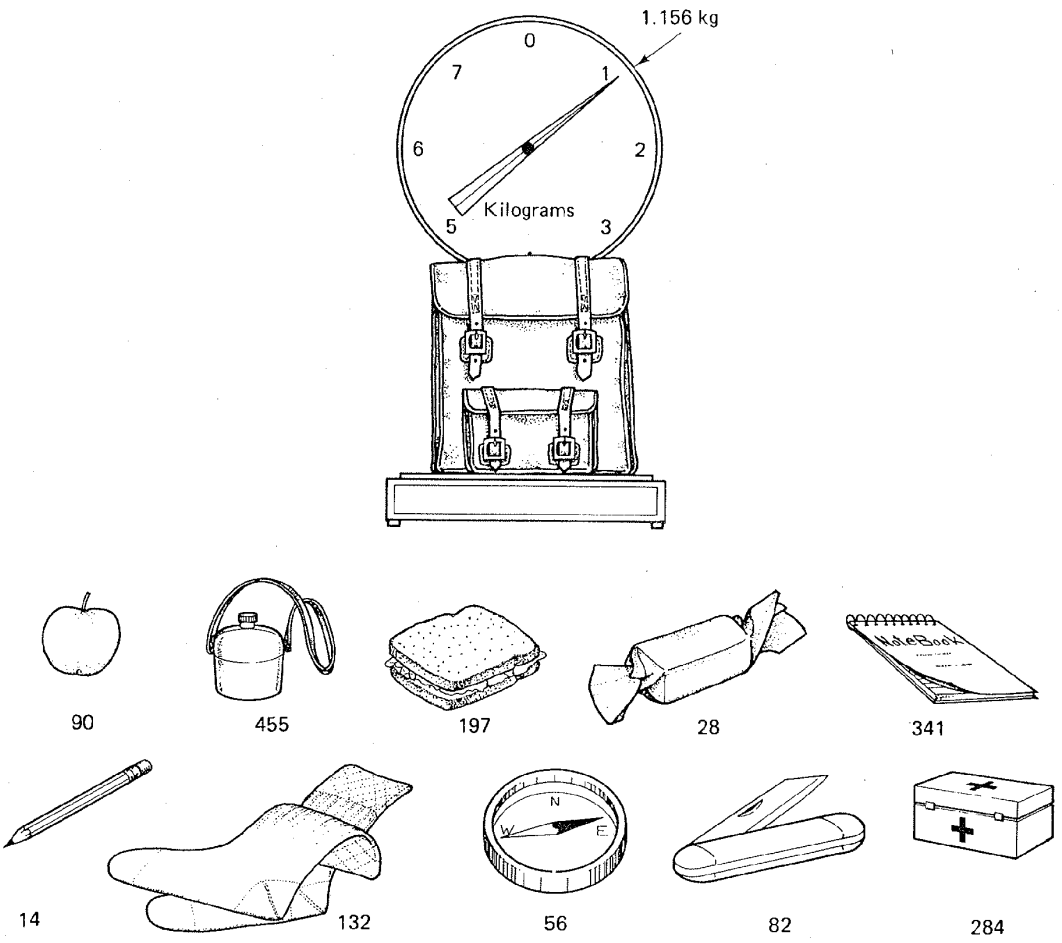


Figure 12.19 Knapsack problem.

The knapsack problem can be stated as follows: Given S and knowing $\mathbf{a}$, determine $\mathbf{x}$.

Example 12.6 Knapsack Example

Given $\mathbf{a} = 1, 2, 4, 8, 16, 32$ and $S = \mathbf{ax} = 26$, find $\mathbf{x}$.

Solution

In this example $\mathbf{x}$ is seen to be the *binary* representation of S. The decimal-to-binary conversion should appear more familiar with $\mathbf{a}$ expressed as $2^0, 2^1, 2^2, 2^3, 2^4, 2^5$. The data vector $\mathbf{x}$ is easily found since $\mathbf{a}$ in this example is *super-increasing*, which means that each component of the n-tuple $\mathbf{a}$ is larger than the sum of the preceding components. That is,

$$a_i > \sum_{j=1}^{i-1} a_j \qquad i = 2, 3, \ldots, n \tag{12.41}$$

When **a** is super-increasing, the solution of **x** is found by starting with $x_n = 1$ if $S \geq a_n$ (otherwise $x_n = 0$), and continuing, as follows:

$$x_i = \begin{cases} 1 & \text{if } S - \sum_{j=i+1}^{n} x_j a_j \geq a_i \\ 0 & \text{otherwise} \end{cases} \tag{12.42}$$

where $i = n - 1, n - 2, \ldots, 1$. From Equation (12.42) it is easy to compute $\mathbf{x} = 0\ 1\ 0\ 1\ 1\ 0$.

Example 12.7 Knapsack Example

Given $\mathbf{a} = 171, 197, 459, 1191, 2410, 4517$ and $\mathbf{S} = \mathbf{ax} = 3798$, find **x**.

Solution

As in Example 12.6, **a** is super-increasing; therefore, we can compute **x** using Equation (12.42), which again yields

$$\mathbf{x} = 0\ 1\ 0\ 1\ 1\ 0$$

12.5.5 A Public Key Cryptosystem Based on a Trapdoor Knapsack

This scheme, also known as the Merkle–Hellman scheme [14], is based on the formation of a knapsack vector that is not super-increasing and is therefore not easy to solve. However, an essential part of this knapsack is a *trapdoor* that enables the authorized user to solve it.

First, we form a super-increasing n-tuple, $\mathbf{a}'$. Then we select a prime number M such that

$$M > \sum_{i=1}^{n} a_i' \tag{12.43}$$

We also select a random number, W, where $1 < W < M$, and we form W^{-1} to satisfy the following relationship:

$$WW^{-1} \text{ modulo-}M = 1 \tag{12.44}$$

The vector $\mathbf{a}'$ and the numbers M, W, and W^{-1} are all kept hidden. Next, we form **a** with the elements from $\mathbf{a}'$, as follows:

$$a_i = Wa_i' \text{ modulo-}M \tag{12.45}$$

The formation of **a** using Equation (12.45) constitutes forming a knapsack vector with a *trapdoor*. When a data vector **x** is to be transmitted, we multiply **x** by **a**, yielding the number S, which is sent on the public channel. Using Equation (12.45), S can be written as follows:

$$S = \mathbf{ax} = \sum_{i=1}^{n} a_i x_i = \sum_{i=i}^{n} (Wa_i' \text{ modulo-}M)x_i \tag{12.46}$$

The authorized user receives S and, using Equation (12.44), converts it to S':

$$
\begin{aligned}
S' = W^{-1}S \text{ modulo-}M &= W^{-1} \sum_{i=1}^{n} (Wa_i' \text{ modulo-}M)x_i \text{ modulo-}M \\
&= \sum_{i=1}^{n} (W^{-1}Wa_i' \text{ modulo-}M)x_i \text{ modulo-}M \\
&= \sum_{i=1}^{n} a_i' x_i \text{ modulo-}M \\
&= \sum_{i=1}^{n} a_i' x_i
\end{aligned}
\tag{12.47}
$$

Since the authorized user knows the secretly held super-increasing vector $\mathbf{a}'$, he or she can use S' to find $\mathbf{x}$.

12.5.5.1 Use of the Merkle–Hellman Scheme

Suppose that user A wants to construct public and private encryption functions. He first considers the super-increasing vector $\mathbf{a}' = (171, 197, 459, 1191, 2410, 4517)$

$$\sum_{i=1}^{6} a_i' = 8945$$

He then chooses a prime number M larger than 8945, a random number W, where $1 \leq W < M$, and calculates W^{-1} to satisfy $WW^{-1} = 1$ modulo-M.

$$\left.\begin{array}{l} \text{Choose } M = 9109 \\ \text{choose } W = 2251 \\ \text{then } W^{-1} = 1388 \end{array}\right\} \text{ kept hidden}$$

He then forms the trapdoor knapsack vector as follows:

$$a_i = a_i' \; 2251 \text{ modulo-}9109$$

$$\mathbf{a} = 2343, 6215, 3892, 2895, 5055, 2123$$

User A makes public the vector $\mathbf{a}$, which is clearly not super-increasing. Suppose that user B wants to send a message to user A.

If $\mathbf{x} = 0\ 1\ 0\ 1\ 1\ 0$ is the message to be transmitted, user B forms

$$S = \mathbf{ax} = 14{,}165 \text{ and transmits it to user } A$$

User A, who receives S, converts it to S':

$$
\begin{aligned}
S' = \mathbf{a}'\mathbf{x} &= W^{-1}S \text{ modulo-}M \\
&= 1388 \cdot 14{,}165 \text{ modulo-}9109 \\
&= 3798
\end{aligned}
$$

Using $S' = 3798$ and the super-increasing vector $\mathbf{a}'$, user A easily solves for $\mathbf{x}$.

Public key schemes are generally too slow for data encryption. With large

encrypted data networks, the biggest problem is how to distribute and manage the keys; the public key systems appear to be very promising for use in key management.

12.6 CONCLUSION

In this chapter we have presented the basic models and goals of the cryptographic process. We looked at some early cipher systems and reviewed the mathematical theory of secret communications established by Shannon. We defined a system that can exhibit perfect secrecy and established that such systems can be implemented but that they are not practical for use where high-volume communications are required. We also considered practical security systems that employ Shannon's techniques known as confusion and diffusion to frustrate the statistical endeavors of a cryptanalyst.

The outgrowth of Shannon's work was utilized by IBM in the LUCIFER system, which later grew into the National Bureau of Standards' Data Encryption Standard (DES). We outlined the DES algorithm in detail. We also considered the use of linear feedback shift registers (LFSR) for stream encryption systems, and demonstrated the intrinsic vulnerability of an LFSR used as a key generator.

Finally, we looked at the novel area of public key cryptosystems and examined two schemes, the Rivest–Shamir–Adelman (RSA) scheme, based on the product of two large prime numbers, and the Merkle–Hellman scheme, based on the classical knapsack problem. The Merkle–Hellman scheme is now considered broken [15], so that today the RSA scheme seems to be the primary means of implementing public key cryptosystems.

REFERENCES

1. Kahn, D., *The Codebreakers*, Macmillan Publishing Company, New York, 1967.
2. Diffie, W., and Hellman, M. E., "Privacy and Authentication: An Introduction to Cryptography," *Proc. IEEE*, vol. 67, no. 3, Mar. 1979, pp. 397–427.
3. Beker, H., and Piper, F., *Cipher Systems*, John Wiley & Sons, Inc., New York, 1982.
4. Denning, D. E. R., *Cryptography and Data Security*, Addison-Wesley Publishing Company, Reading, Mass., 1982.
5. Shannon, C. E., "Communication Theory of Secrecy Systems," *Bell Syst. Tech. J.*, vol. 28, Oct. 1949, pp. 656–715.
6. Hellman, M. E., "An Extension of the Shannon Theory Approach to Cryptography," *IEEE Trans. Inf. Theory*, vol. IT23, May 1978, pp. 289–294.
7. Smith, J. L., "The Design of Lucifer, a Cryptographic Device for Data Communications," *IBM Research Rep. RC-3326*, 1971.
8. Feistel, H. "Cryptography and Computer Privacy," *Sci. Am.*, vol. 228, no. 5, May 1973, pp. 15–23.

9. National Bureau of Standards, "Data Encryption Standard," *Federal Information Processing Standard (FIPS)*, Publication no. 46, Jan. 1977.
10. United States Senate Select Committee on Intelligence, "Unclassified Summary: Involvement of NSA in the Development of the Data Encryption Standard," *IEEE Commun. Soc. Mag.*, vol. 16, no. 6, Nov. 1978, pp. 53–55.
11. Diffie, W., and Hellman, M. E., "New Directions in Cryptography," *IEEE Trans. Inf. Theory*, vol. IT22, Nov. 1976, pp. 644–654.
12. Rivest, R. L., Shamir, A., and Adelman, L., "On Digital Signatures and Public Key Cryptosystems," *Commun. ACM*, vol. 21, Feb. 1978, pp. 120–126.
13. Knuth, D. E., *The Art of Computer Programming*, Vol. 2, *Seminumerical Algorithms*, 2nd ed., Addison-Wesley Publishing Company, Reading, Mass., 1981.
14. Merkle, R. C., and Hellman, M. E., "Hiding Information and Signatures in Trap-Door Knapsacks," *IEEE Trans. Inf. Theory*, vol. IT24, Sept. 1978, pp. 525–530.
15. Shamir, A., "A Polynomial Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem," *IEEE 23rd Ann. Symp. Found. Comput. Sci.*, 1982, pp. 145–153.

PROBLEMS

12.1. Let X be an integer variable represented with 64 bits. The probability is $\frac{1}{2}$ that X is in the range $(0, 2^{16} - 1)$, the probability is $\frac{1}{4}$ that X is in the range $(2^{16}, 2^{32} - 1)$, and the probability is $\frac{1}{4}$ that X is in the range $(2^{32}, 2^{64} - 1)$. Within each range the values are equally likely. Compute the entropy of X.

12.2. A set of equally likely weather messages are: sunny (S), cloudy (C), light rain (L), and heavy rain (H). Given the added information concerning the time of day (morning or afternoon), the probabilities change as follows:

$$\text{Morning:} \quad P(\text{S}) = \tfrac{1}{8},\ P(\text{C}) = \tfrac{1}{8},\ P(\text{L}) = \tfrac{3}{8},\ P(\text{H}) = \tfrac{3}{8}$$

$$\text{Afternoon:} \quad P(\text{S}) = \tfrac{3}{8},\ P(\text{C}) = \tfrac{3}{8},\ P(\text{L}) = \tfrac{1}{8},\ P(\text{H}) = \tfrac{1}{8}$$

(a) Find the entropy of the weather message.
(b) Find the entropy of the message conditioned on the time of day.

12.3. The Hawaiian alphabet has only 12 letters—the vowels, a, e, i, o, u, and the consonants, h, k, l, m, n, p, w. Assume that each vowel occurs with probability 0.116, and that each consonant occurs with probability 0.06. Also assume that the average number of *information bits* per letter is the same as that for the English language. Calculate the unicity distance for an encrypted Hawaiian message if the key sequence consists of a random permutation of the 12-letter alphabet.

12.4. Estimate the unicity distance for an English language encryption system that uses a key sequence made up of 10 random alphabetic characters:
(a) Where each key character can be any one of the 26 letters of the alphabet (duplicates are allowed).
(b) Where the key characters may not have any duplicates.

12.5. Repeat Problem 12.4 for the case where the key sequence is made up of ten integers randomly chosen from the set of numbers 0 to 999.

12.6. (a) Find the unicity distance for a DES system which encrypts 64-bit blocks (eight alphabetic characters) using a 56-bit key.

(b) What is the effect on the unicity distance in part (a) if the key is increased to 128 bits?

12.7. In Figures 12.8 and 12.9, P-boxes and S-boxes alternate. Is this arrangement any more secure than if all the P-boxes were first grouped together, followed by all the S-boxes similarly grouped together? Justify your answer.

12.8. What is the output of the first iteration of the DES algorithm when the plaintext and the key are each made up of zero sequences?

12.9. Consider the 10-bit plaintext sequence 0 1 0 1 1 0 1 0 0 1 and its corresponding ciphertext sequence 0 1 1 1 0 1 1 0 1 0, where the rightmost bit is the earliest bit. Describe the five-stage linear feedback shift register (LFSR) that produced the key sequence and show the initial state of the register. Is the output sequence of maximal length?

12.10. Following the RSA algorithm and parameters in Example 12.5, compute the encryption key, e, when the decryption key is chosen to be 151.

12.11. Given e and d that satisfy ed modulo-$\phi(n) = 1$, and a message that is encoded as an integer number, M, in the range $(0, n - 1)$ such that the gcd $(M, n) = 1$. Prove that $(M^e \text{ modulo-}n)^d \text{ modulo-}n = M$.

12.12. Use the RSA scheme to encrypt the message $M = 3$. Use the prime numbers $p = 5$ and $q = 7$. Choose the decryption key, d, to be 11, and calculate the value of the encryption key, e.

12.13. Consider the following for the RSA scheme.

(a) If the prime numbers are $p = 7$ and $q = 11$, list five allowable values for the decryption key, d.

(b) If the prime numbers are $p = 13$, $q = 31$, and the decryption key is $d = 37$, find the encryption key, e, and describe how you would use it to encrypt the word "DIGITAL."

12.14. Use the Merkle–Hellman public key scheme with the super-increasing vector, $\mathbf{a}' = 1, 3, 5, 10, 20$. Use the following additional paramcters: a large prime number $M = 51$ and a random number $W = 37$.

(a) Find the nonsuper-increasing vector, $\mathbf{a}$, to be made public, and encrypt the data vector 1 1 0 1 1.

(b) Show the steps by which an authorized receiver decrypts the ciphertext.

DIGITAL COMMUNICATIONS

Fundamentals and Applications

BERNARD SKLAR

Readers of this upper-level book will find comprehensive coverage of digital communication systems and an assortment of signal processing techniques that have arisen over the past two decades.

Bernard Sklar's treatment of this rapidly growing field not only provides the reader with organization and structure but also ensures awareness of the *"big picture"* even while delving into the details. Throughout the book, emphasis is placed on digital communications and also on those analog fundamentals that are important to the development of digital theory.

Key content features include:

- Modulation and Error-Correction Coding
- What the Link Budget Tells the System Engineer
- Spread Spectrum and Encryption Techniques
- Multiplexing and Multiple Access
- Synchronization

P T R Prentice Hall
Englewood Cliffs, New Jersey 07632