Spread spectrum and Pseudonoise Sequences

The notes in this document are partly based on Chapter 7 of the textbook.

1 Overview

A fundamental problem in wireless communications is to allow multiple share simultaneously a finite amount of spectrum. This is referred to as the *multiple access* problem. This problem arises in wired environments as well, e.g., the Ethernet. At a high level, multiple access schemes can be divided into five categories:

- Frequency division multiple access (FDMA): In FDMA, the spectrum is divided into individual channels, each of which is allocated to an individual user for a period of time. During this period no other user can use the same channel (piece of the spectrum); other users, however, could be simultaneously using other disjoint pieces of the spectrum.
- Time division multiple access (TDMA): In TDMA, time is divided into slots, and in any slot one user transmits/receives over the entire spectrum for the duration of the slot. No other user uses the channel during the slot. Data has to be sent in bursts, so digital data and digital modulation must be used with TDMA.
- Spread Spectrum multiple access (SSMA): In spread spectrum, the transmission bandwidth of the signals is several orders of magnitude greater than the minimum bandwidth required for transmitting a given piece of information. However, many users can share the same spread spectrum bandwidth without interfering with one another. Spread spectrum uses encoding techniques that are different than the ones we discussed in the preceding chapter; these will be the focus of this chapter. The two most common techniques for implementing SSMA are *frequency-hopping spread spectrum* (FHSS) and *direct sequence spread spectrum* (DSSS).
- Space division multiple access (SDMA): Another multiple access method is to divide the space into areas such that a receiving node can distinguish among signals received from these areas. In SDMA, a node can simultaneously receive/send signals to/from multiple users, spread across its 360-degree span, by using directional antennas that concentrate energy within a narrow arc of the span.
- Carrier sense multiple access (CSMA): In CSMA, wireless nodes sense the medium for an ongoing transmission, to avoid collisions. CSMA schemes widely used in ad hoc networks as well as the IEEE 802.11 wireless LAN, both of which will be studied in detail later in the course.

2 Frequency-hopping spread spectrum

A signal is broadcast over a pseudo-random sequence of frequencies, a sequence obtained by a pseudo-random generator. The pseudo-random sequence is referred to as the *spreading code* or a *pseudo-noise* sequence. The energy of the signal is equally divided among different frequencies.

An FHSS system consists of a modulator that uses a standard modulation technique to produce a signal centered around some base frequency. This signal is next modulated again to produce a signal centered at a different frequency, which is determined by the pseudo-noise sequence. This second modulation is done using a chip signal $c(t)$ which characterizes the pseudo-noise sequence. The spacing between the carrier frequencies used in the chip sequence is usually close to the bandwidth of the input signal. The interval of time that an FHSS system spends at a hop frequency is referred to as the *chip duration*. As we will see shortly, this is a parameter that is independent of the data rate of the original signal and can be varied to obtain different FHSS schemes.

FHSS using BFSK. Consider using BFSK as the first data modulation scheme. The output of the first modulation step is the signal

$$s_d(t) = A\cos\left(2\pi(f_0 + 0.5(b_i+1)\Delta f)t\right), \text{ for } iT < t < (i+1)T,$$

where f_0 is the base frequency, Δf is the frequency separator in the BFSK scheme, b_i is the ith bit, and T is the duration of a single bit.

Now, suppose the next frequency as determined by the pseudo-noise sequence is f_i. We obtain a product signal of

$$\begin{aligned} p(t) &= A\cos\left(2\pi(f_0 + 0.5(b_i+1)\Delta f)t\right)\cos(2\pi f_i t) \\ &= 0.5A\left(\cos(2\pi(f_0 + 0.5(b_i+1)\Delta f + f_i)t) + \cos(2\pi(f_0 + 0.5(b_i+1)\Delta f - f_i)t)\right). \end{aligned}$$

We can eliminate the second half of the above sum and obtain signal with frequency centered around $f_0 + f_i$. When the signal is received at the receiver, the original bit value b_i can be recovered by following the reverse process, as long as the sender and the receiver are synchronized with respect to the pseudo-noise sequence.

FHSS using MFSK. A natural generalization of the above is to use multi-level signaling technique such as MFSK. If the MFSK modulated signal is given by $s_d(t) = A\cos(2\pi f_i t)$, where $f_i = f_c + (2i-1-M)f_d$ is the frequency used for the ith signal element, f_c is the center frequency, and M is the number of signal elements. This MFSK signal is then translated to a new frequency every T_c seconds by modulating it with the FHSS carrier.

For a data rate R, the duration of a bit $T = 1/R$, so the duration of a signal element is $T_s = \log_2 MT$. If $T_c \geq T_s$, the system is referred to as a slow FHSS; otherwise, it is a fast FHSS. In FHSS, the bandwidth of a hop is $W_d = MF_d$, where M is the number of different signal elements. The total transmission bandwidth is $W_s = NW_d$ if N is the number of different hop-channels.

FHSS is used in one of the RF interfaces for IEEE 802.11, in combination with two- or four-level Gaussian FSK.

FHSS-CDMA. In an FHSS/CDMA system, each user has its own hopping sequence. In such a system, interference occurs only when two users land on the same frequency at the same time. If the selected codes are random and independent, then the collision probability can be made very small. Codes can also be selected and synchronized so as to eliminate multiple-user interference. FHSS-based CDMA is adopted in Bluetooth.

Performance of FHSS. Consider a noise/interference jammer of the same bandwidth and fixed power S_j on the signal carrier frequency. We have a ratio of E_b/N_j equal to $E_b W_d/S_j$. The processing gain is given by $W_s/W_d = N$.

3 Direct sequence spread spectrum

In DSSS, each bit of the digital data is mapped to a sequence of bits in the transmitted signal, using a spreading code.

DSSS using BPSK. We first modulate the digital data using BPSK to obtain

$$s_d(t) = Ad(t)\cos(2\pi f_c t),$$

where f_c is the carrier frequency.

Let $c(t)$ denote the chipping signal. The spread spectrum modulation is obtained by taking a product of $s_d(t)$ and $c(t)$ to get $s(t) = Ad(t)c(t)\cos(2\pi f_c t)$. At the receiver, the reverse operation is performed. The chipping signal satisfies the condition $(c(t))^2 = 1$ so that the receiver is able to decode the encoded signal.

If jamming signal $S_j(t)$ is a sinusoid with the same frequency as the center frequency, then $S_j(t) = \sqrt{2S_j}\cos(2\pi f_c t)$

The processing gain G is given by $T/T_c = W_s/W_d$.

4 Code division multiple access (CDMA)

Code division multiple access is a multiplexing technique that allows multiple users to simultaneously access a shared medium. Each user A is assigned an n-bit spreading code c_A that is orthogonal with the code of any other current user (for a suitable choice of n). Each "bit" of c_A is a 1 or a -1. The encoding is simple: a 1 translates to c_A, while a 0 translates to $-c_A$.

On receiving a digital signal, the receiver decodes any message from A by simply taking a dot-product of the signal (n symbol periods long) with A's spreading code as follows.

$$S_A(d) = \sum_i d_i c_i^A.$$

If ith bit is 1, then $S_A(d) = n$ and $-n$ otherwise.

In CDMA, we ideally select codes such for any other user B, $S_A(c_B) = 0$ (orthogonality). So, if A and B simultaneously transmit bits b_A and b_B (in $\{0, -1\}$), respectively, the combined signal is $b_A c_A + b_B c_B$. When the decoding takes place, we obtain

$$\begin{aligned} S_A(b_A c_A + b_B c_B) &= b_A S_A(c_A) + b_B S_B(c_B) \\ &= b_A. \end{aligned}$$

Multiple users can now simultaneously access the channel by simply encoding the data using their own spreading code and having the "interference" canceled by the orthogonality of the codes.

There are several important factors to discuss in connection with the workings of CDMA. First, it is clear that one does not need strict orthogonality; approximate orthogonality will suffice as long as it clearly separates out the different user data. In fact, guaranteeing strict orthogonality is also not practically feasible. Second, synchronization of the users to the chipping process is important for the orthogonality condition to be useful; one can settle for limited sychronization if the codes are approximately orthogonal, even with small shifts, but this may limit the number of users. Finally, some kind of power control is required to ensure that the strength of a signal is not overwhelmed by the strengths of other signals. A problem arising due to lack of power control is the "near/far problem": the signal of a user that is far from a base station is dominated by that from a nearby user, both transmitting at the same signal strengths, since the attenuation of the first signal is much more than that of the second. As a result, when the combined signal is decoded using the code of the farther user (and both the codes are not perfectly orthogonal to each other), the decoded value is not correct.

5 Pseudo-noise Sequences

Linear feedback shift register (LFSR). LFSRs are one of the simplest ways to generate pseudo-random sequences. In an LFSR, any bit is determined by a linear combination of the previous n bits, for a suitable choice of n. In particular, we may have an LFSR in which

$$B_n = A_0 B_0 \oplus A_1 B_1 \oplus A_2 B_2 \oplus \ldots \oplus A_{n-1} B_{n-1}.$$

Since any bit is a function of the previous n bits, we can show that every LFSR has a period of $2^n - 1$. Consider any window of n bits. As we slide $2^n - 1$ times forward, we cover $2^n - 1$ n-bit strings. If any two of the strings repeat, then we already have the period at most $2^n - 1$. Furthermore, since the all 0s cannot be generated (unless the initial value is all 0s in which all values remain all 0s), within 2^n steps at least one window is identical to another n-bit window encountered earlier in the process; this the period of the sequence is $2^n - 1$.

M-Sequences. M-sequences, or maximal length sequences, are pseudonoise sequences generated by LFSR that have maximum period. That is, any sequence that is generated by an n-bit LFSR and has period $2^n - 1$ is an m-sequence. An m sequence has several desirable properties. It has an almost uniform distribution of 0s and 1s. Specifically, in any window of $2^n - 1$ bits, it has 2^{n-1} ones and $2^{n-1} - 1$ zeros. Moreover, in any window of $2^n - 1$ bits, it has one run of ones of length n, one run of zeros of length $n - 1$, and in general, 2^{n-i-2} runs of ones and 2^{n-i-2} runs of zeros of length i, $1 \leq i \leq n - 2$.

For spread spectrum applications, one important measure of the "randomness" of a pseudonoise sequence is its autocorrelation function, which gives the correlation of a period of the sequence with its cyclic shifts. The smaller the correlation, the easier is it for the sender-receiver pair to synchronize; furthermore, interference effects are also somewhat alleviated. For a sequence $B_1, \ldots, B_N$ of ± 1, the periodic autocorrelation function R is given by

$$R(\tau) = \frac{1}{N} \sum_{k=1}^{N} B_k B_{k-\tau}.$$

It can be shown that the periodic autocorrelation of an m-sequence is

$$R(\tau) = \begin{cases} 1 & \tau = 0, N, 2N, \ldots \\ -\frac{1}{N} & \text{otherwise} \end{cases}$$

In spread spectrum applications, when the codes for different users are generated from different sources, then a similar measure of interest is the cross-correlation of the two codes. The cross-correlation function of two sequences $A_1, \ldots, A_n$ and $B_1, \ldots, B_N$ of ± 1s is defined as

$$R_{A,B}(\tau) = \frac{1}{N} \sum_{k=1}^{N} A_k B_{k-\tau}.$$

It is intuitive that we would like the cross-correlation function of two user spread spectrum codes to be low since that would allow a receiver to discriminate among spread spectrum signals generated by different m-sequences.

Gold sequences. For DSSS systems, we need to generate seperate codes for different users. A Gold sequence set provides such a collection of codes. It is obtained by repeatedly taking bit-wise XORs of two "uncorrelated" sequences. The starting sequence is an m-sequence a. The second sequence a' is obtained by sampling in a deterministic manner – every qth bit, for some appropriate q – from multiple copies of a until the length of a' is the same as a. The ith subsequent sequence is obtained by a bit-wise XOR of a with a i-bit shifted copy of a'. In order for the Gold sequences to satisfy desirable properties – such as each code being an m-sequence – a and q have to satisfy certain conditions, which are described in the text.

Orthogonal codes. Orthogonal codes are a set of sequences whose all pairwise cross-correlations are zero. Walsh codes are the most common orthogonal codes and are used in CDMA applications. The set of Walsh codes of length $n = 2^k$, $k \geq 0$, consists of the n rows of an $n \times n$ matrix, which is recursively defined as follows.

$$W_1 = (0) \qquad\qquad W_{2n} = \begin{pmatrix} W_n & W_n \\ W_n & \overline{W_n} \end{pmatrix},$$

where $\overline{A}$ is the matrix A with every bit replaced by its logical NOT. We now prove that codes in a $2^n \times 2^n$ Walsh matrix are orthogonal to each other for any integer $n \geq 1$. The proof is by induction on n. The claim is trivially true for the base case $n = 1$. For the induction step, let us assume that the claim is true for a $2^{n-1} \times 2^{n-1}$ Walsh matrix. Consider any two distinct row vectors w_i and w_j of a $2^n \times 2^n$ Walsh matrix, where we replace 0s by -1s. We consider three cases.

The first case is when $i, j \leq 2^{n-1}$. In this case, $w_i \cdot w_j$ is the sum of two quantities, each of which is the dot-product of the ith and jth row vectors of the $2^{n-1} \times 2^{n-1}$ Walsh matrix. By the induction hypothesis, both these quantities are zero, thus establishing the induction step for this case.

The second case is when $i, j > 2^{n-1}$. In this case, $w_i \cdot w_j$ is the sum of two quantities, the first of which is the dot-product of row $i - 2^{n-1}$ and row $j - 2^{n-1}$ of the $2^{n-1} \times 2^{n-1}$ Walsh matrix, and the second is the corresponding dot-product of the complement matrix. By the induction hypothesis, both these quantities are zero, thus establishing the induction step for this case.

The final case is when $i \leq 2^{n-1}$ and $j > 2^{n-1}$. Again, the dot-product is the sum of two quantities, the first being the dot-product of row i and row $j - 2^{n-1}$ of the $2^{n-1} \times 2^{n-1}$ Walsh matrix, while the other being the dot-product of row i of the $2^{n-1} \times 2^{n-1}$ Walsh matrix and row $j - 2^{n-1}$ of the complement of the $2^{n-1} \times 2^{n-1}$ Walsh matrix. Since these quantities complement each other, we obtain a dot-product of 0, thus completing the proof.