Lecture 16

Today: (1) Spread Spectrum

- OH Fri 9-noon (J. Kemp).
- Tuesday: Guest lecture by Prof. Sneha Kasera on wireless networking. I will be out of town Mon-Tue, March 15-16 for a conference.

1 Spread Spectrum

“Spread spectrum” is the use of a much wider bandwidth than necessary in a radio communications system, in order to achieve other objectives.

For example, frequency modulation (FM) historically was considered to be widely spread in spectrum (broadcast FM radio uses 200 kHz to send audio, which is inherently a 3-10 kHz bandwidth signal). However, FM was used because it could mitigate the effects of multipath fading, and could achieve better signal quality at low SNR.

The two major types of spread spectrum modulation are:

1. Frequency Hopping Spread Spectrum (FH-SS)
2. Direct-Sequence Spread Spectrum (DS-SS)

Both types of SS use pseudo-random codes, periodic sequences of zeros and ones. These are not actually random – a computer generates them using simple binary feedback logic. But, an average person would look at the sequence and judge them to be random. So they are called pseudo-random.

1.1 FH-SS

FH-SS pseudo-randomly changes center frequency each “hopping period”, $T_h$. Bluetooth is a FH-SS system, which achieves a (coded) bit rate of 1 Mbps (potentially up to 3 Mbps), but uses 80 MHz of spectrum, in 79 different center frequencies, with a hopping period $T_h = 1/1600$ s/hop. While at each center frequency, the modulation is similar to things we have been talking about, e.g., FSK, DPSK, QPSK, etc.

Three benefits of FH-SS are:
1. **Interference avoidance**: There may be significant interference at a few of the center frequencies. But even if we totally lose all bits during those hops, we will be able to recover using the bits received during successful (non-interfered) hops. We also avoid being an interferer to someone else’s signal for too long.

2. **Multiple Access**: Two devices can occupy the same spectrum and operate without coordinating medium access at all. Their transmissions will “collide” some small fraction of the time, but not often enough to cause failure.

3. **Stealth**: There is an advantage to switching randomly among frequencies when an eavesdropper doesn’t know your hopping pattern – they will not be able to easily follow your signal. This was the original reason for the discovery and use of FH-SS (by actor and inventor Hedy Lamarr, in 1940 [?]).

### 1.2 DS-SS

We will see that DS-SS has the same three advantages as FH-SS, but for different reasons.

Direct-sequence spread spectrum simply uses a pulse shape that has a wide bandwidth. This pulse shape is known as a pseudo-noise signal, which is, essentially, itself a BPSK-modulated signal, but which “pseudo-random” data that is known to both transmitter and receiver.

![Figure 1: Pseudo-noise ("Barker") code used in 802.11b DS-SS modulation.](image)

Figure 1 shows the PN signal used in 802.11b. Can you draw on top of Figure 1 what the pulse shape would be if it were not DS-SS, and instead, simply modulated BPSK, with a SRRC pulse shape? You can see that the bandwidth of the DS-SS signal will be very large compared to that of the BPSK signal.

We can describe the modulated signal, sent by user $k$, as (from Rappaport):

$$s_k(t) = a_k(t)p_k(t)\cos(2\pi f_c t)$$
where $a_k(t)$ is the bit signal, +1 or -1, which indicates the data bits to be sent. For example, the Barker code uses bits $[+1, 1, +1, 1, +1, +1, +1, 1, 1]$. As before, we have a pulse shape $p_k(t)$, however, in this case, $p_k(t)$ is not simply a SRRC pulse. It is a high bandwidth pseudo-noise signal. Essentially, $p_k(t)$ is a BPSK-modulated signal itself. The “bits” of the PN signal are called “chips” to distinguish them from bits. We denote the number of chips in $p_k(t)$ as $PG$, it is the number of chips per bit. It is called PG because it is also called the processing gain. The period of each chip is denoted $T_c$, so

$$PG = \frac{T_s}{T_c} = \frac{R_c}{R_s}$$

Where $R_c = 1/T_c$ is the chip rate. For example, 802.11b has a chip rate of 11 M (chips per second) and a symbol rate of 1 M (symbols per second). As another example, in IS-95, the “short code” has $PG = 2^{15}+1 = 32768$.

The bandwidth of a DS-SS symbol, when chips have the SRRC shape, is

$$B = (1 + \alpha)R_c$$

Which is then $PG$ times the bandwidth of the BPSK signal would have been as a narrowband signal.

Recall that the SNR required to demodulate a signal is given by:

$$SNR = \frac{\mathcal{E}_b}{N_0 B}$$

So with DS-SS, the SNR is lowered by a factor of $PG$

$$SNR = \frac{\mathcal{E}_b}{N_0 (1 + \alpha)R_c} = \frac{\mathcal{E}_b}{N_0 (1 + \alpha)PG}$$

However, If you thought this signal was just a plain-old BPSK signal, i.e., didn’t know the PN signal, you’d need the regular SNR $\frac{\mathcal{E}_b}{N_0 (1 + \alpha)}$, which is $PG$ times higher. This makes us understand advantage #3 of DS-SS: Stealth. Knowing the PN signal allows one to demodulate the signal with $PG$ times less SNR than an eavesdropper could. If the $PG$ is high enough, the signal would be extremely difficult to detect at all, but could still be used by the intended receiver.

Advantage #1: Reception of DS-SS uses the same principle as discussed earlier – the received signal is correlated with the known PN signal. What if a narrowband interference signal was also in the received signal? Well, this narrowband signal would effectively be spread, when it is multiplied by the PN signal in the receiver. In contrast, the desired signal is de-spread (becomes narrowband again) due to correlation with the PN signal. The spread interference can then be partially filtered out using a narrowband filter. See Figure 6.50 in Rappaport (page 333).
Advantage #2: Further, DS-SS can be designed for some particular benefits for multiple access. These relate to the near-orthogonality of the particular PN codes used in DS-SS. In short, some sets of PN signals are nearly orthogonal or completely orthogonal to each other; and some PN signals have the property that the PN signal is orthogonal to itself at a different time delay.

First, consider sets of PN signals orthogonal to each other. One example is the set of Gold codes (used in the GPS network). Gold code signals are nearly orthogonal with each other.

Another example is the Walsh-Hadamard sequence set, used in IS-95 (CDMA). The WH-64 sequences, shown in Figure 2(c), are used in IS-95. The 64 WH signals are exactly orthogonal to each other. These signals provide a means on the downlink to send 64 simultaneous user signals, and yet have each mobile perform a correlation operation that completely zeros out the signal sent from the BS to all 63 other mobiles. When one mobile correlates with its signal, $p_k(t)$, it has zero correlation with the other 63 WH signals.

![Figure 2](image)

Figure 2: Walsh Hadamard sequences are in the rows of the above images. The signal is +1 during black squares, and -1 during red squares: (a) two WH-2 sequences in two rows, (b) four WH-4 sequences in four rows, and (c) sixty-four WH-64 sequences in 64 rows.

Note that WH sequences are not used on the uplink, because they are only orthogonal if time-synchronized. Mobiles aren’t able to time-synchronize with each other very well.

Second, consider the autocorrelation of a PN signal (the correlation of a signal with itself at different time delays). The autocorrelation is defined as

$$R_p(\tau) = \int_0^{T_s} a_k(t)p_k(t)p_k(t - \tau)dt$$

First assume that the data signal $a_k(t) = 1$. If $\tau = 0$, the value of
$R_p(\tau)$ is simply the energy in the PN signal $p_k(t)$ over a duration $T_s$. For $\tau$ a multiple of $T_s$, that is, the period of $p_k(t)$, we get the same value, i.e., $R_p(nT_s) = R_p(0)$. For in between values of $\tau$, (say, $T_c < \tau < T_s - T_c$), PN signals have a nearly (but not quite) zero autocorrelation. Generally, for these $\tau$, $R_p(\tau) \approx -R_p(0)/PG$. Figure 4(top plot) shows an example for the Barker code used in 802.11b.

Now, if the data was modulated, the exact $-R_p(0)/PG$ is gone, but we still do have $|R_p(\tau)| \leq R_p(0)/PG$. Notice that the autocorrelation becomes strongly negative when the bit sent was a -1.

![Autocorrelation of "Barker" code used in 802.11b DS-SS modulation, (a) with all symbol values = +1, that is, $a_k(t) = 1$, and (b) with data modulation.](image)

**Example: PN code generation**

Consider Figure 4(a). Let each state of $s_1, \ldots, s_4$ to be either zero or one (but not all zeros to start out). At each stage, one bit is output, and there is mod 2 addition performed to fill $s_1$. Let’s calculate the output.
Figure 4: Linear feedback shift register (LFSR) generation of an (a) 4-stage, or (b) 5-stage, maximal-length PN code; and (c) the generated code sequence for the 4-stage LFSR.