**TELCOM 2720 Wireless Communications**

Fundamental Concepts in Wireless Systems (continued)

## Frequency Hopping

Traditionally in wireless systems the transmitter/recevier pair communicate on a fixed frequency band for the duration of the communication session. This is illustrated in the (a) part of the top figure on page 1 of the fourth class handout. As we saw in the discussion of radio propagation models the interference, noise and fading properties change somewhat with the frequency band used. In order to ensure fairness (in terms of the quality of the channel) and combat noise and fading some wireless systems employ frequency hopping. In frequency hopping systems the the transmitter/receiver pair regularly jumps or hops from one frequency band to another throughout the session. This is illustrated in the (b) part of the top figure on page 1 of the class handout. Frequency hopping was orginally invented to prevent jamming and eavesdropping in military analog radio systems, but is now used in digital cellular systems and wireless LAN's. Both the transmitter and receiver must know the hopping pattern employed which consists of an algorithm to determine the order in which the frequency bands are hopped and the dwell time (i.e., time duration a given band is used) at each frequency band. Frequency hopping systems are also termed frequency hopped spread spectrum (FHSS) systems, since the user spreads its data across a wider spectrum than just one channel. Frequency hopped systems are classified into two categories: slow hopped and fast hopped. In slow hopping systems the dwell time at each frequency band is long enough to transmitt multiple bits, where as in fast hopped systems multiple frequency hops occur for each bit transmitted. Fast hopping is primarily used in military systems.

## Adaptive Equalizers

In digital wireless systems, one method used to combat the poor quality of the radio channel is the use of adaptive equalizers at receivers as shown in the figure on page 2 of the fourth class handout. Equalizers are used in the demodulation process to correctly interpret the transmitted symbols (i.e., whether a 1 or 0 was sent) and essentially act as a inverse of the channel distortion. The primary effect of adaptive equalization is to remove intersymbol interference due to multipath and movement effects. The equalizer has two modes: training and operation. In the training mode a fixed length known bit pattern is sent by the transmitter to the receiver; at the receiver the known bit pattern is used to adjust the parameters of the equalizer to correctly determine the bit pattern. When the training is complete the equalizer has estimated the effects of the radio channel and is ready to switch modes to operation where it interprets the transmitted user symbol data. The figure on page 3 of the handout shows a block diagram of a typical adaptive equalizer - which in this case is just a linear tap delay line filter. Since the wireless channel propagation/fading effects vary with location, movement and time the training sequence must be repeated regularly for the equalizer to track the changing channel characeristics. Thus some portion of the bit rate of the channel is used exclusively for this purpose.

**Antenna Diversity**

Antenna diversity (also called space diversity) is a common technique used in wireless systems to combat fast fading and to a lesser extent shadowing. The concept is based on the random propagation effects of the wireless channel (remember shadowing typically has a lognormal distribution and fast fading a Rayleigh distribution). Thus two receiving antenna spaced several wavelengths apart are unlikely to have the received signal in a fade at exactly the same instant. This is illustrated in the bottom figure on page 4 of the handout where, two recieving antenna are utilized at the base station. The fading effects of each antenna should be relatively independent and the base station instantaneously selects the strongest signal from the two antenna as the received signal as shown in the figure.

Typically the spacing between the antenna is in the range of 10 to 20 wavelengths and this results in a 3 to 6 dB lessening of the fading. Of course such a setup is impractical for the mobile terminal and diversity is applied primarily at base stations.

**Error Control Coding**

One of the basic techniques used in *digital* wireless systems to improve the channel quality is the use of error control coding. The basic idea is to add redundant information to the data bit stream to in order to detect or correct transmission errors. The addition of this redundant information will of course reduce the effective data rate of the channel. The two strategies possible in error control coding are error detection and forward error correction.

*Error detection*

In error detection the basic idea is to add redundant bits which allows one to determine if one or more transmission errors has occured. When an error is detected the errored data is discarded and the receiver must request retransmission of the data. Typically one adds a fixed number $r$ of the redundant bits (often called parity bits) to a block of k data bits to detect errors and the resulting transmitted block is $n = k + r$ bits long. At the receiver, it takes the received $k$ data bits recomputes the $r$ parity bits and compares them with the received parity bits to see if correct reception has occurred. If an error is detected, the receiver requests a repeat transmission of the entire block of $n$ bits and repeats the detection procedure.

Example 1: Consider a simple error detection strategy where for every 7 data bits one parity bit is added and the value of the parity bit is used to enforce even parity (that is make the number of '1's in the 8 bit block always an even number). Let the transmitted word have the format $p_0 d_6 d_5 d_4 d_3 d_2 d_1 d_0$ where, $d_i$ is the $ith$ data bit and $p_0$ is the single parity bit. Then if the user data is for example 1011001 which has even parity then $p_0 = 0$; whereas if the data bits are 1101110 which has odd parity that results in $p_0 = 1$. To illustrate how this strategy can be used to detect errors consider the first case above where $p_0 d_6 d_5 d_4 d_3 d_2 d_1 d_0 = 01011001$ if this is transmitted and received with one transmission error say in $d_5$ resulting in a received byte of 01111001 which has odd parity the reciever knows that an error has occured and requests retransmission of the byte.

*Forward error correction*

The idea behind forward error correction is to map the data stream into a code word stream which allows one to detect and correct errors. There are two basic types of error correcting codes: block and convolutional. In block codes, parity bits are added to blocks of data bits to make code words. In a block encoder, $k$ information bits are encoded into $n$ code bits by the appending of $r = n - k$ redundant parity bits. The block code is referred to as a $(n, k)$ code and the rate (effective data rate) of the code is defined as $k/n$. The ability of the block code to detect and correct errors is a function of the *coding distance*.

The coding distance between two equal length words is measured by the number of bit positions in which the two words differ. Let $d[x, y]$ denote the coding distance between words $x$ and $y$ then $d[000; 111] = 3$; whereas $d[010; 011] = 1$.

Example 2: Consider the 8 code words $[0000; 0011; 0101; 0110; 1001; 1010; 1100; 1111]$ the minimum coding distance between any pair of words is 2.

A simple example of a block code is the $(7, 4)$ Hamming code in which 3 additional parity bits are added for every 4 data bits. $[p_0\ p_1\ p_2\ d_3\ d_2\ d_1\ d_0]$. This code is given on page 5 of the handout. The Hamming code $(7, 4)$ has the minimum coding distance 3 between any pair of words. A single bit error in a transmitted code word will result in the received seven bits being distance 1 from the correct code word and distance 2 from all other code words. Thus one can *correct* all single bit errors in a code word by simply decoding any received word as it's closest distance code word. One can show that the $(7, 4)$ Hamming code can detect errors in 2 or fewer bits in a code word and correct a single bit error.

In order to see the advantages of using forward error correction, one must consider the probability of error in blocks of data. We assume that bit errors occur according to independent Bernoulli trials that is on each bit transmission we perform a coin toss and with probability $p$ an error occurs and with probability $q = (1 - p)$ no error occurs. The value of $p$ is called the bit error rate (BER) of the channel. For the case when the 4 data bits are sent with no error correction used, if the BER $=0.01$, the probability of getting a block of 4 bits correct is $(1 - .01)^4 = 0.9606$. Therefore, the probability of error in transmission of a block is almost 0.04. On the other hand, if we use the (7, 4) block code the probability of getting correct transmission is the probability of all seven bits being correctly received plus the probability of having only 1 bit error in the block of 7 bits. This is equivalent to $(1-0.01)^7 + \binom{7}{6}(0.01)(1-0.01)^6 = 0.9321 + 0.0659 = 0.9980$. Therefore, the probability of error in a block is decreased to 0.002, which is an improvement of one order of magnitude in the error rate (typical of error correcting codes). However, the effective data rate is degraded to $k/n\times$ the channel rate.

In cellular systems convolutional codes are often used since they have fast coding/decoding properties. Convolutional codes operate on a data stream using a sliding window and produce a continuous stream of output encoded bits. The size $j$ of the sliding window is termed the constraint length of the code. For each data stream bit the encoder looks over the last $j$ bits and produces $m$ output bits, the resulting rate $r$ of the code is $r = 1/m$. The number of errors that can be detected are a function of $j$ and $m$. A basic convolutional coder is shown on page 6 of the handout. At the receiver the last $j$ encoded bits are bufferred and decoded to reproduce the data stream. The decoding process is complex and can be shown to be equivalent to finding the min cost path through a trellis graph.

*Interleaving*

Both error detection and error correction codes work on the principal the errors occur randomly and independently. However, in wireless channels several factors combine (shadowing, Rayleigh fading) to make the errors occur in bursts. That is a sequence of several bits in error will occur. Interleaving is often used to overcome this problem. The basic idea in interleaving is to make the errors that occur in clusters on the channel appear randomly spread out over a longer time interval. This is accomplished by performing permutation of the error control encoded data at the transmitter. Receivers perform the reverse permutation (de-interleaving) operation to place the bits into their orginal sequence.

For example consider the (7,4) Hamming block code which can correct one error in the block of seven bits. where three consecutive code words are buffered. Let bits of the $i$ th block of encoded data be denoted $b_7^i, b_6^i, b_5^i, b_4^i, b_3^i, b_2^i, b_1^i$ In the normal case the encoded data bits of the three buffered blocks are transmitted in sequential order (from right to left) ( $b_7^3, b_6^3, b_5^3, b_4^3, b_3^3, b_2^3, b_1^3;\ b_7^2, b_6^2, b_5^2, b_4^2, b_3^2, b_2^2, b_1^2;\ b_7^1, b_6^1, b_5^1, b_4^1, b_3^1, b_2^1, b_1^1$ ) . In contrast when they are interleaved the 21 bits are transmitted in groups of three with only one bit in each group from the same 7 bit block. That is the transmitted sequence is (from right to left) ( $b_7^3, b_7^2, b_7^1, b_6^3, b_6^2, b_6^1, b_5^3;\ b_5^2, b_5^1, b_4^3, b_4^2, b_4^1, b_3^3, b_3^2;\ b_3^1, b_2^3, b_2^2, b_2^1, b_1^3, b_1^2, b_1^1$). Using this interleaving a sequence of three bit errors can be tolerated, since when the three consecutive bits in error are de-interleaved they will be in three different blocks. Resulting in 1 bit error in each block which can be corrected with the Hamming block code.