ECE461: Digital Communications

Lecture 6: Rate Efficient Reliable Communication

Introduction

We now move to rate efficient reliable communication (energy efficiency tends to come for free in this scenario). In this lecture we see that there are block communication schemes smarter than the naive repetition coding seen earlier that promise arbitrarily reliable communication while still having a non-zero data rate. We begin by setting the stage for studying rate efficient reliable communication by carefully dividing the transmitter strategy of mapping the information bits to transmit voltages into two distinct parts:

1. maps information bits into *coded* bits by adding more redundancy: the number of coded bits is larger than the number of information bits and the ratio is called the *coding rate*. This process is generally called *coding*.

2. map coded bits directly into transmit voltages. This is done *sequentially*: for instance, if only two transmit voltages are allowed ($\pm\sqrt{E}$) then every coded bit is sequentially mapped into one transmit voltage. If four possible transmit voltages are allowed ($\pm\sqrt{E}, \pm\frac{\sqrt{E}}{3}$), then every two consecutive coded bits are mapped into a single transmit voltage sequentially. This mapping is typically called *modulation* and can be viewed as a labeling of the discrete transmit voltages with a binary sequence.

The receiver could also be potentially broken down into two similar steps, but in this lecture we will continue to focus on the ML receiver which maps the received voltages *directly* into information bits. Focusing on a simple binary modulation scheme and the ML receiver, we see in this lecture that there are plenty of good coding schemes: in fact, we will see that *most* coding schemes promise arbitrarily reliable communication provided they are decoded using the corresponding ML receiver!

Transmitter Design: Coding and Modulation

We are working with an energy constraint of E, so the transmit voltage is restricted to be within $\pm\sqrt{E}$ at each time instant. For simplicity let us restrict that only two transmit voltages are possible: $+\sqrt{E}$ and $-\sqrt{E}$.[1]

If we are using T time instants to communicate, this means that the number of *coded* bits is T, one per each time instant. With a coding rate of R, the number of *information* bits (the size of the data packet) is $B = RT$. Surely, $R \leq 1$ in this case. The scenario of $R = 1$ exactly corresponds to the *sequential* communication scheme studied in Lecture 4. As we saw there, the reliability level approaches zero for large packet sizes. The point is that even though we have spaced the transmit voltages far enough apart (the spacing is $2\sqrt{E}$ in this case), the chance that at least one of the bits is decoded incorrectly approaches unity when there are a lot of bits. The idea of introducing redundancy between the number of information bits and coded bits (by choosing $R < 1$) is to ameliorate exactly this problem.

[1] We will explore the implications of this restriction in a couple of lectures from now.

Linear Coding

As we have seen, coding is an operation that maps a sequence of bits (information bits, specifically) to a *longer* sequence of bits (the coded bits). While there are many types of such mappings, the simplest one is the *linear code*. This is best represented mathematically by a matrix C whose elements are drawn from $\{0, 1\}$:

$$\begin{bmatrix} \text{vector of} \\ \text{coded bits} \end{bmatrix} = C \begin{bmatrix} \text{vector of} \\ \text{information} \\ \text{bits} \end{bmatrix}. \tag{1}$$

Here the vector space operations are all done on the binary field $\{0, 1\}$: i.e., multiplication and addition in the usual modulo 2 fashion. The dimension of the matrix C is $T \times RT$ and it maps a vector of dimension $RT \times 1$ (the sequence of information bits) into a vector of dimension $T \times 1$ (the sequence of coded bits). The key problem is to pick the appropriate code C such that the unreliability with ML decoding at the receiver is arbitrarily small. In this lecture we will see that *almost all* matrices C actually have this property!

A Novel Approach

To study this we will consider the set $\mathcal{C}$ of all possible binary matrices C: there are 2^{RT^2} number of them (each entry of the matrix can be 0 or 1 and there are RT^2 entries in the matrix). We will show that the average unreliability, averaged over all the matrices C:

$$\overline{\mathbb{P}[\mathcal{E}]} \stackrel{\text{def}}{=} \frac{1}{2^{RT^2}} \sum_{C \in \mathcal{C}} \mathbb{P}[\mathcal{E}|C], \tag{2}$$

is arbitrarily small for large packet sizes B (and hence large time T). In Equation (2) we have used the notation $\mathbb{P}[\mathcal{E}|C]$ to denote the unreliability of communication with the appropriate ML receiver over the AWGN channel *when using the code* C at the transmitter.[2] If $\overline{\mathbb{P}[\mathcal{E}]}$ is arbitrarily small, then most code matrices C must have an error probability that is also arbitrarily small. In fact, only at most a polynominal (in RT) number of codes can have poor reliability.

Calculating Average Unreliability

This unreliability level is the average unreliability experienced, averaged over all possible information bit sequences:

$$\mathbb{P}[\mathcal{E}|C] = \frac{1}{2^{RT}} \sum_{k=1}^{2^{RT}} \mathbb{P}[\mathcal{E}|B_k, C], \tag{3}$$

where B_k is the kth information packet B (there are $2^B = 2^{RT}$ possible information packets). The error event $\mathcal{E}$ occurs when the likelihood of the T received voltages is larger for some

[2] Keep in mind that the ML receiver will, of course, depend on the code C used at the transmitter.

other packet B_j with $j \neq k$. The probability of this event will depend on the nature of the code C and is, in general, quite complicated to write down precisely. As in the previous lecture, we will use the union bound to get an upper bound on this unreliability level:

$$\mathbb{P}\left[\mathcal{E}|B_k, C\right] < \sum_{j\neq k, j=1}^{2^{RT}} \mathbb{P}_2\left[B_k, B_j|C\right], \tag{4}$$

where we have denoted $\mathbb{P}_2\left[B_k, B_j|C\right]$ as the probability of mistakenly concluding that B_j is the information packet when actually B_k was transmitted using the code C.

Substituting Equations (4) and (3) into Equation (2) we get

$$\overline{\mathbb{P}\left[\mathcal{E}\right]} < \frac{1}{2^{RT}} \sum_{k=1}^{2^{RT}} \sum_{j\neq k, j=1}^{2^{RT}} \left(\frac{1}{2^{RT^2}} \sum_{C\in\mathcal{C}} \mathbb{P}_2\left[B_k, B_j|C\right] \right). \tag{5}$$

The Key Calculation

We now come to the key step in this whole argument: *evaluating* the value of the expression

$$\frac{1}{2^{RT^2}} \sum_{C\in\mathcal{C}} \mathbb{P}_2\left[B_k, B_j|C\right]. \tag{6}$$

From our derivation in Lecture 4, we know that the error probability

$$\mathbb{P}_2\left[B_k, B_j|C\right] = Q\left(\frac{d}{2\sigma}\right) \tag{7}$$

where d is the Euclidean distance between the vectors of transmitted voltages corresponding to the information packets B_k, B_j using the code C. Since we have only a binary transmit voltage choice, the distance d simply depends on the number of time instants where the *coded bits* corresponding to the information packets B_k, B_j using the code C are *different*. Suppose the coded bits are different at ℓ of the possible T locations. Then the Euclidean distance squared

$$d^2 = \ell 4E. \tag{8}$$

The idea is that for each time where the coded bits are different, the square of the distance is $4E$ (since the corresponding transmit voltages are $\pm\sqrt{E}$ leading to a distance of $2\sqrt{E}$). So we can write the key expression in Equation (6) using Equations (7) and (8) as

$$\frac{1}{2^{RT^2}} \sum_{C\in\mathcal{C}} \mathbb{P}_2\left[B_k, B_j|C\right] = \sum_{\ell=0}^{T} f_\ell^{(k,j)} Q\left(\frac{\sqrt{\ell E}}{\sigma}\right). \tag{9}$$

Here we have denoted $f_\ell^{(k,j)}$ as the fraction of the codes that lead to exactly ℓ mismatches in the T coded bits for the pair of information packets B_k and B_j.

While the value of ℓ depends on the code C and the pair of packets B_k, B_j, we can calculate $f_\ell^{(k,j)}$ readily. In fact, it is the *same* for all pair of information packets B_k, B_j.

Towards this calculation, let us focus on the coded bit at any specific time instant, say the first one. Since we are considering all possible codes C, there is no special bias towards this bit being a 1 or a 0. In other words, the fraction of codes that result in this coded bit being 1 (or 0) is exactly 0.5. This is true for every information packet. Furthermore, there is no special bias for this coded bit to be the same or different for any pair of information packets (B_k and B_j in our notation). In other words, the fraction of codes for which the coded bit of interest (say, the first one out of the total T) is the same for any pair of information packets B_k, B_j is exactly 0.5. So, the fraction of codes that lead to exactly ℓ mismatches in the T coded bits for any pair of information packets B_k, B_j is

$$f_\ell^{(k,j)} = \binom{T}{\ell} \left(\frac{1}{2}\right)^T. \tag{10}$$

The fraction is the same for all pairs (k, j). To elaborate a bit more on how we arrived at the expression in Equation (10), we see that:

1. there are $\binom{T}{\ell}$ possible ways of choosing the ℓ mismatches in a sequence of T coded bits;

2. the term $\left(\frac{1}{2}\right)^T$ is simply a normalization term – the sum of all the fractions of codes that lead to exactly ℓ mismatches, summed over all ℓ, should be unity:

$$\sum_{\ell=0}^{T} f_\ell^{(k,j)} = 1. \tag{11}$$

   Using the binomial formula

$$\sum_{\ell=0}^{T} \binom{T}{\ell} = 2^T \tag{12}$$

   we see that the normalization term has to be $\left(\frac{1}{2}\right)^T$.

Now substituting Equation (10) in Equation (9) we arrive at

$$\frac{1}{2^{RT^2}} \sum_{C \in \mathcal{C}} \mathbb{P}_2 \left[B_k, B_j | C\right] = \sum_{\ell=0}^{T} \binom{T}{\ell} \left(\frac{1}{2}\right)^T Q\left(\frac{\sqrt{\ell E}}{\sigma}\right). \tag{13}$$

We can use the upper bound on the $Q(\cdot)$ function

$$Q(a) \leq \frac{1}{2} e^{\frac{-a^2}{2}}, \quad a \geq 0, \tag{14}$$

in Equation (13) to write

$$\frac{1}{2^{RT^2}} \sum_{C \in \mathcal{C}} \mathbb{P}_2 \left[B_k, B_j | C\right] < \sum_{\ell=0}^{T} \binom{T}{\ell} \left(\frac{1}{2}\right)^T e^{-\frac{\ell E}{2\sigma^2}}. \tag{15}$$

Using the binominal formula

$$\sum_{\ell=0}^{T} \binom{T}{\ell} e^{-\frac{\ell E}{2\sigma^2}} = \left(1 + e^{-\frac{E}{2\sigma^2}}\right)^T \tag{16}$$

in Equation (15) we come to

$$\begin{aligned} \frac{1}{2^{RT^2}} \sum_{C \in \mathcal{C}} \mathbb{P}_2 \left[B_k, B_j | C\right] &< \left(1 + e^{-\frac{E}{2\sigma^2}}\right)^T \left(\frac{1}{2}\right)^T && (17) \\ &= 2^{T \log_2 \left(1 + e^{-\frac{E}{2\sigma^2}}\right)} \left(\frac{1}{2}\right)^T && (18) \\ &= 2^{-TR^*} && (19) \end{aligned}$$

where we have defined

$$\begin{aligned} R^* &\stackrel{\text{def}}{=} \log_2 \left(\frac{2}{1 + e^{-\frac{E}{2\sigma^2}}}\right) && (20) \\ &= 1 - \log_2 \left(1 + e^{-\frac{E}{2\sigma^2}}\right). && (21) \end{aligned}$$

Reliable Rate of Communication

The denouement is near now. Substituting Equation (19) into Equation (5) we see the average unreliability, averaged over all possible linear codes C is no more than

$$\begin{aligned} \overline{\mathbb{P}\left[\mathcal{E}\right]} &< \frac{1}{2^{RT}} \sum_{k=1}^{2^{RT}} \sum_{j \neq k, j=1}^{2^{RT}} 2^{-R^* T} && (22) \\ &< 2^{RT} 2^{-R^* T}. && (23) \end{aligned}$$

As long as the data rate R is strictly less than R^* (defined in Equation (21)), the average unreliability level becomes arbitrarily small for large T (and hence large information packet sizes). The term R^* is a simple function of the SNR (defined, as usual, to be the ratio of E to σ^2) and is strictly positive. It can be interpreted as the threshold below which arbitrary reliable communication data rates are possible. Further more, this performance is guaranteed by many linear codes C when coupled with the appropriate ML decoder at the receiver end.

Finally, any linear code C that guarantees reliable communication is also energy efficient. This is explored in a homework exercise.

Looking Ahead

In this lecture we focused primarily on the design of the transmitter so that we can communicate reliably while still being rate efficient. We supposed the availability of the corresponding ML decoder at the receiver end. The ML decoder involves finding that information packet

yielding the largest likelihood of the T received voltages. We have to calculate the likelihood for each possible information packet, 2^{RT} of them: the number of such choices grows exponentially with the number of information bits RT. Supposing a constant computation complexity to calculate the likelihood the total computation complexity of the ML receiver grows exponentially with the size of the information packet. Even with moderate sized data packets (say 1000s of bits) this is just too expensive (in fact, widely considered to be *impossible*) to implement in digital circuitry. In the language of CS (computer science) we say that the ML decoder algorithm is NP (non-polynomial time).

To complete the story of rate efficient reliable communication, we turn to simplifying the receiver design. While the ML decoder is indeed the optimal receiver, there are suboptimal decoders that perform about just as well in practice. This is the focus of the next lecture.