Modern Coding Theory

Instructor: Giuseppe Caire

Copyright G. Caire
Lecture 1: What is Channel Coding?
Communication Protocol Stack

- **App (http)**
- **Transp (TCP/UDP)**
- **Networks (IP)**
- **MAC**
- **PHY**

---

Copyright G. Caire
Copyright G. Caire
Where things may go bad?

- App (http)
- Transp (TCP/UDP)
- Networks (IP)
- MAC
- PHY

Copyright G. Caire
Physical layer

\[ b \rightarrow \text{Enc/Mod} \rightarrow x(t) \rightarrow \text{Channel} \rightarrow y(t) \rightarrow \text{Dem/Dec} \rightarrow \hat{b} \]
Linear distortion and additive noise

\[ b \rightarrow \text{Enc/Mod} \rightarrow x(t) \rightarrow H(f) \rightarrow y(t) \rightarrow \text{Dem/Dec} \rightarrow \hat{b} \]

\[ y(t) = x(t) * h(t) + z(t) \]
• 50% of “Communication theory” is dedicated to the mathematical machinery (signal representation) and algorithms for transforming the actual waveform channel into a vector channel defined over $\mathbb{C}^n$. 

$$y = Hx + z$$
The other 50% is dedicated to methods for coding and modulation on the Gaussian vector channel.
Separate demodulation and decoding
Separate demodulation and decoding

\[ \begin{align*}
01 & \rightarrow \text{Enc} \rightarrow \text{Mod} \rightarrow H \rightarrow \text{Dem} \rightarrow \hat{u} \\
11 & \rightarrow \hat{b} \\
10 & \rightarrow \hat{b}
\end{align*} \]
- Binary symmetric channel (BSC).
• Packet Erasure Channel (PEC). ..... the BEC is a special case.
Coding for the PEC

- In order to protect packets we need to send groups of packets (blocks) “together”.

- Consider \( (b_1, b_2, \ldots, b_k) \) and represent each packet \( b_i \) as a symbol from a finite field.

- For example, if \( b_i \in \{0, 1\}^q \), then we can identify it with an element of \( \mathbb{F}_{2^q} \).

- Linear encoding:

\[
(u_1, u_2, \ldots, u_n) = (b_1, b_2, \ldots, b_k) \cdot \begin{bmatrix}
g_{11} & g_{12} & \cdots & g_{1n} \\
g_{21} & g_{22} & \cdots & g_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
g_{k1} & g_{k2} & \cdots & g_{kn}
\end{bmatrix}
\]
Coding for the PEC

• More compactly:

\[ u = bG \]

\[
\begin{array}{ccc}
    u_1 & u_2 & u_3 \\
    \hline
    \end{array} = \begin{array}{cc}
    b_1 & b_2 \\
    \end{array} \begin{array}{ccc}
    1 & 0 & 1 \\
    0 & 1 & 1 \\
    \end{array}
\]
The received sequence of coded packets is given by

\[ v_i = \begin{cases} u_i & \text{if no erasure} \\ ? & \text{if erasure} \end{cases} \]

Example:

\[
\begin{array}{c}
\begin{array}{c}
\text{u}_1 \quad \text{u}_3
\end{array}
\end{array}
= \begin{array}{c}
\begin{array}{c}
\text{b}_1 \quad \text{b}_2
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
1 \\ 0 \\ 1 \\ 1
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
0 \\ 1 \\ 1
\end{array}
\end{array}
\end{array}
\]
The received sequence of coded packets is given by

\[ v_i = \begin{cases} u_i & \text{if no erasure} \\ ? & \text{if erasure} \end{cases} \]

Example:

\[
\begin{align*}
\text{u}_1 & \quad \text{u}_3 \\
\text{b}_1 & \quad \text{b}_2 \\
\end{align*}
\]

\[
\begin{pmatrix}
1 \\
0 \\
1 \\
\end{pmatrix}
\]
Decoding: solving a linear system

\[ \begin{bmatrix} u_1 & u_3 \end{bmatrix} = \begin{bmatrix} b_1 & b_2 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \]
Decoding: solving a linear system

\[
A^{-1} \begin{bmatrix} u_1 & u_3 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}
\]
Tanner graph

\[ \begin{array}{c}
  \text{u}_1 \\
  \text{u}_2 \\
  \text{u}_3 \\
  + \\
\end{array} \]
Generalization: MDS codes

- A \((n, k)\)-MDS code is a linear code over \(\mathbb{F}_Q\) with the following property: the transmitted codeword is uniquely identified by any set of \(k\) symbols.

- It follows that a \((n, k)\)-MDS code can correct any pattern of at most \(n - k\) erasures.

- Code rate and channel capacity:

\[
R = \frac{k}{n} = 1 - \frac{n - k}{n}
\]

- For very large \(n\), the average number of erasures per block is \(en\). We require

\[
en < n - k = n(1 - R) \Rightarrow R < 1 - e = C
\]
Coding for noisy channels: BSC

- Consider the BSC with transition probability $p < 1/2$.
- Pack information bits into blocks of $k$.
- Append to each block $m = n - k$ “parity” symbols.
- **Decoding:** Given the received word $y$, find the “most likely” codeword $x \in C$. 
Hamming code \((n = 7, k = 4)\)

\[
C = \left\{ (0, 0, 0, 0, 0, 0, 0), (1, 0, 0, 0, 1, 1, 0), (0, 1, 0, 0, 0, 1, 1), (1, 1, 0, 0, 1, 0, 1), (0, 0, 1, 0, 1, 1, 1), (1, 0, 1, 0, 0, 0, 1), (0, 1, 1, 0, 1, 0, 0), (1, 1, 1, 0, 0, 1, 0), (0, 0, 0, 1, 1, 0, 1), (1, 0, 0, 1, 0, 1, 1), (0, 1, 0, 1, 1, 1, 0), (1, 1, 0, 1, 0, 0, 0), (0, 0, 1, 1, 0, 1, 0), (1, 0, 1, 1, 0, 0, 0), (0, 1, 1, 1, 0, 1), (1, 1, 1, 1, 0, 1), (1, 1, 1, 1, 1, 1) \right\}
\]
Hamming code \((n = 7, k = 4)\)

- Generator matrix form:

\[
(x_1, x_2, x_3, x_4, x_5, x_6, x_7) = (b_1, b_2, b_3, b_4) \begin{bmatrix}
1 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 \\
\end{bmatrix}
\]

- The code as the row-space of \(G\) (subspace of the vector space \(F_2^7\)):

\[
C = \left\{ x \in F_2^7 : x = bG, \quad b \in F_2^4 \right\}
\]
Hamming code \( (n = 7, k = 4) \)

- Parity-check matrix form:

\[
\begin{pmatrix}
1 & 1 & 0 \\
1 & 0 & 1 \\
0 & 1 & 1 \\
1 & 1 & 1 \\
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
(x_1, x_2, x_3, x_4, x_5, x_6, x_7)
\end{pmatrix} = (0, 0, 0)
\]

- The code as the null-space of \( H^T \) (subspace of the vector space \( \mathbb{F}_2^7 \)):

\[
C = \left\{ x \in \mathbb{F}_2^7 : xH^T = 0 \right\}
\]
Minimum Hamming distance

- Hamming weight: $w_H(x) =$ number of non-zero symbols.

- Hamming distance: $d_H(x, x') = w_H(x - x')$.

- For a linear code with parity-check matrix $H$, $d_{\text{min}} = \min_{x \neq x'} d_H(x, x')$ is given by the minimum number $d$ for which there exists a set of $d$ linearly dependent columns in $H$.

\[
H = \begin{bmatrix}
1 & 1 & 0 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 1
\end{bmatrix}
\]

- For the Hamming $(7, 4)$-code we have $d_{\text{min}} = 3$. 
Communication channel models

- Discrete Memoryless Channel: \( \{\mathcal{X}, P_{Y|X}, \mathcal{Y}\} \) such that

\[
P(Y = y|X = x) = \prod_{i=1}^{n} P_{Y|X}(y_i|x_i)
\]

- Continuous-output Memoryless Channels: \( \mathcal{Y} \) is \( \mathbb{R} \) or \( \mathbb{C} \) (ref: recall communication theory), \( \mathcal{X} \) may be either \( \mathbb{R}, \mathbb{C} \) or a discrete set of points embedded in \( \mathbb{R} \) or \( \mathbb{C} \). conditional pdf \( p_{Y|X}(y|x) \) such that

\[
p_{Y|X}(y|x) = \prod_{i=1}^{n} p_{Y|X}(y_i|x_i),
\]

where \( p_{Y|X}(y|x) \) is the conditional density of the output given the input.
Basic definitions

- A code $C$ over $X$ is a subset of $X^n$. $n$ is called block length, $|C| = 2^{nR}$ is the size of the code, $R = \frac{1}{n} \log_2 |C|$ is the rate of the code, measured in bits per symbol.

- An encoder is an indexing function that maps information messages to codewords. Without loss of generality, we can assume that information messages are binary $k$-tuples, where $k = nR$. Therefore, an encoder for $C$ is a mapping

$$f : \{0, 1\}^k \rightarrow C$$

- A decoder is a mapping of the possible channel outputs $Y^n$ onto the information messages, that is,

$$g : Y^n \rightarrow \{0, 1\}^k$$
MAP decoding

- The average probability of error for the code $C$ is defined as

$$P_e(C) = 2^{-nR} \sum_{x \in C} \mathbb{P}(g(Y) \neq x|X = x)$$

- Optimal decoding rule (minimizing $P_e(C)$): we write

$$P_e(C) = 1 - 2^{-nR} \sum_{x \in C} \mathbb{P}(g(Y) = x|X = x)$$

$$= 1 - 2^{-nR} \sum_{x \in C} \sum_{y \in \mathcal{R}_x} P_{Y|X}(y|x)$$

where $\mathcal{R}_x = \{y \in \mathcal{Y}^n : g(y) = x\}$ is the decision region of the codeword $x$. 
• The optimal decision rule has decision regions:

\[ \mathcal{R}_x = \{ y \in \mathcal{Y}^n : P_{Y|X}(y|x) \geq P_{Y|X}(y|x') \ \forall \ x' \in \mathcal{C} \} \]

(ties can be arbitrarily broken ... e.g., by randomization on the boundary).

• Using Bayes rule we have

\[ P_{Y|X}(y|x)2^{-nR} = P(Y = y, X = x) = P(X = x|Y = y)P(Y = y) \]

• This yields the Maximum A-posteriori Probability (MAP) decision rule:

\[ \mathcal{R}_x = \{ y \in \mathcal{Y}^n : P(X = x|Y = y) \geq P(X = x'|Y = y) \ \forall \ x' \in \mathcal{C} \} \]
For the BSC, we have

\[ P_{Y|X}(y|x) = \prod_{i=1}^{n} p^{w_H(y_i-x_i)}(1 - p)^{1- w_H(y_i-x_i)} = p^{d_H(x,y)}(1 - p)^{n-d_H(x,y)} \]

The MAP decoding rule is given by:

\[ \hat{x} = \arg \max_{x \in \mathcal{C}} \quad p^{d_H(x,y)}(1 - p)^{n-d_H(x,y)} = \arg \min_{x \in \mathcal{C}} \quad d_H(x, y) \]

for \( p \in [0, 1/2] \) this is the (minimum Hamming distance rule).
Min-distance decoding
Upper and lower bounds to $P_e$

- In general, computing the error probability for a given codeword $x \in C$ is a hard task
  \[ P_e(x) = 1 - \sum_{y \in \mathcal{R}_x} P_{Y|X}(y|x) \]
  (the sum becomes an integral for a continuous-output channel).

- We can find simpler and general upper and lower bounds to $P_e(C) = 2^{-nR} \sum_{x \in C} P_e(x)$ resulting from the MAP decision rule.

- For any given pair of codewords $x' \neq x$, we define the pairwise error event
  \[ \{x \rightarrow x'\} \triangleq \{y \in \mathcal{Y}^m : P_{Y|X}(y|x) \leq P_{Y|X}(y|x')\} \]
  and the pairwise error probability (PEP) as
  \[ P(x \rightarrow x') \triangleq \mathbb{P}(\{x \rightarrow x'\}|X = x) \]
Then, we have

\[
P_e = 2^{-nR} \sum_{x \in C} P_e(x)
\]

\[
= 2^{-nR} \sum_{x \in C} \mathbb{P} \left( \bigcup_{x' \neq x} \{x \rightarrow x'\} \mid X = x \right)
\]

\[
\leq 2^{-nR} \sum_{x \in C} \sum_{x' \neq x} \mathbb{P} (\{x \rightarrow x'\} \mid X = x)
\]

\[
\leq 2^{-nR} \sum_{x \in C} \sum_{x' \neq x} P (x \rightarrow x')
\]
Next, we consider a lower bound on the error probability:

\[ P_e = 2^{-nR} \sum_{x \in C} P_e(x) \]

\[ = 2^{-nR} \sum_{x \in C} \mathbb{P} \left( \bigcup_{x' \neq x} \{x \rightarrow x'\} \bigg| X = x \right) \]

\[ \geq 2^{-nR} \sum_{x \in C} \max_{x' \neq x} \mathbb{P} \left( \{x \rightarrow x'\} \bigg| X = x \right) \]

\[ = 2^{-nR} \sum_{x \in C} \max_{x' \neq x} P(x \rightarrow x') \]

E.g., for the BSC, \( \max_{x' \neq x} P(x \rightarrow x') \) is achieved when \( x' \) is a nearest neighbor of \( x \).
End of Lecture 1