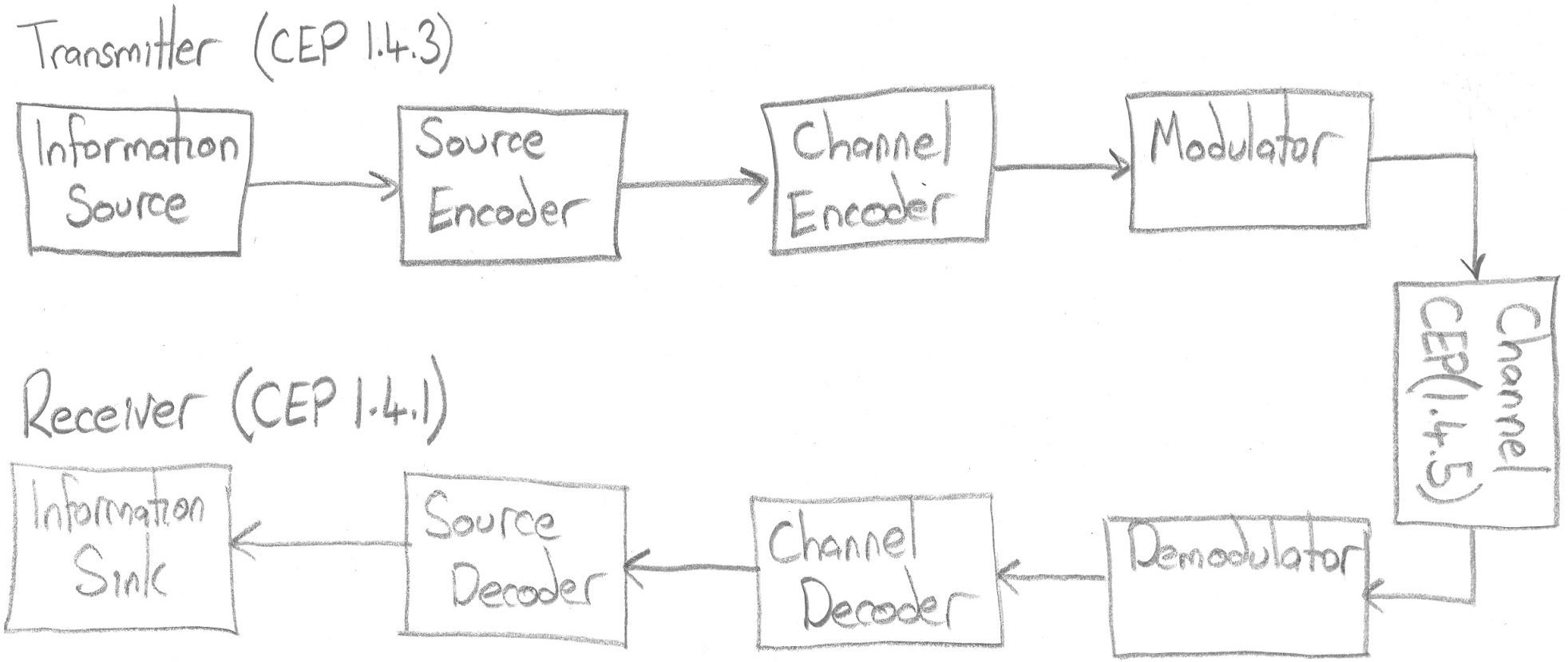
ELEC1323 Communications

10 Channel Coding

Communication schemes



Sources of noise (CEP 9.3)

When transmitting a signal over a channel, it is typically subject to:

- Interference, caused by the transmissions of other communicating devices.
- Sky noise, caused by the emission of radio waves by warm bodies in space, the atmosphere and on the surface of the Earth.
- Johnson noise, caused by the thermal agitation of electrons within the transmitter and receiver circuits.
- Shot noise, caused by random fluctuations in the number of electrons that flow as a current within the transmitter and receiver circuits.
- Quantum noise, caused by random fluctuations in the number of photons that flow in an optical fibre.
- Flicker and partition noise, caused by semiconductors.

Additive White Gaussian Noise (CEP 9.4 – 9.5)

• All sources of noise can be thought of as being combined into a single source of Additive White Gaussian Noise (AWGN).

• Suppose we have a digitally modulated signal $v(t)$, which was obtained using FSK, ASK, PSK or digital-QAM, for example.

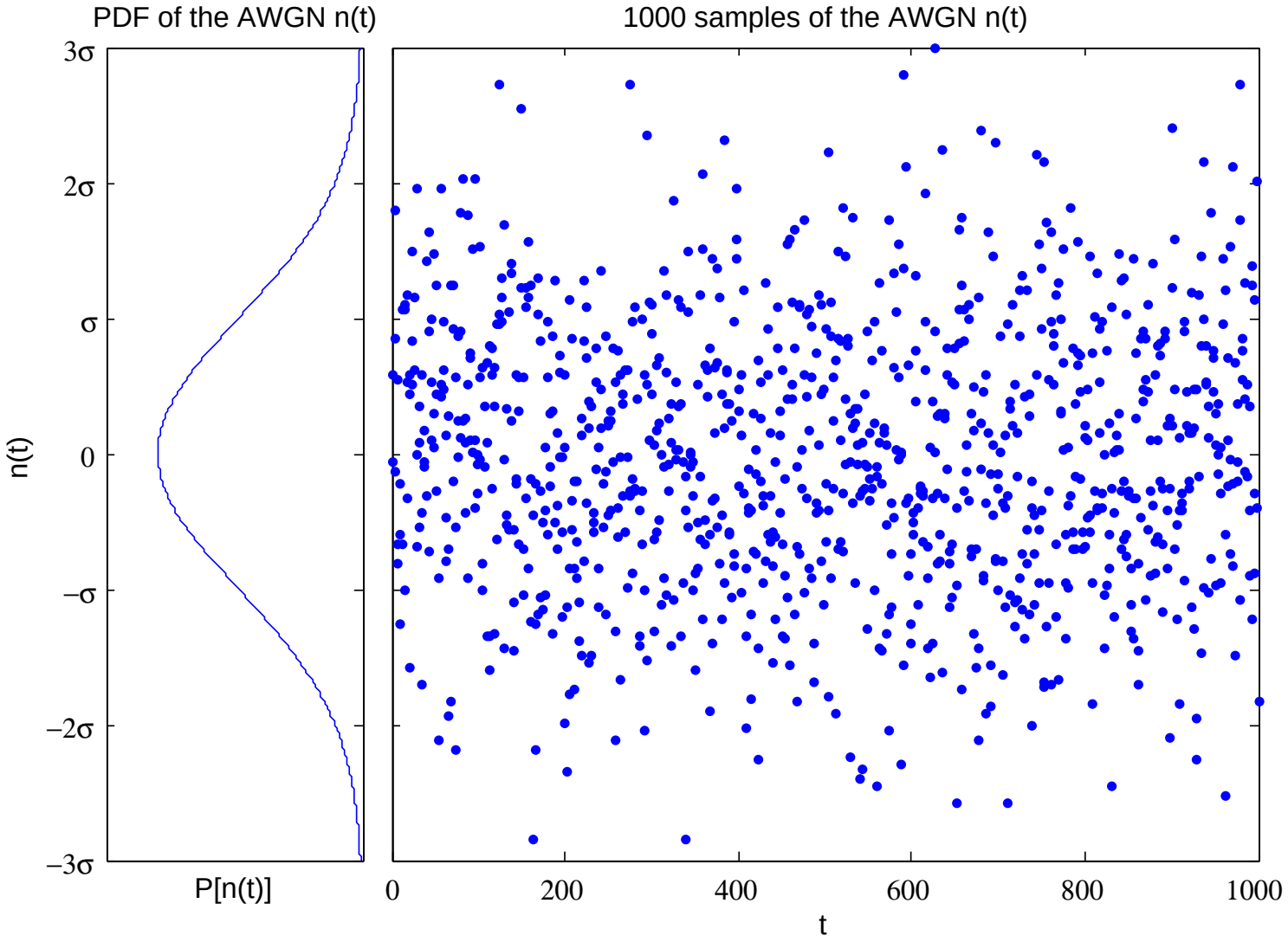
• If we transmit this over a channel that introduces AWGN, the received signal will be corrupted according to $\hat{v}(t) = v(t) + n(t)$, where $n(t)$ is the AWGN.

• The AWGN $n(t)$ is additive because it is added to the signal $v(t)$.

• The AWGN $n(t)$ is white because its amplitude spectrum is flat within the bandwidth of the signal $v(t)$ (white light has a flat amplitude spectrum within the visible part of the electromagnetic spectrum).

• The AWGN $n(t)$ is Gaussian because its value at a particular time t resembles a random value chosen from a Gaussian distribution having a mean of 0 and a standard deviation of σ.

Additive White Gaussian Noise (CEP 9.3 – 9.5)



Bit errors

- Noise in the received signal $\hat{v}(t)$ can cause the demodulator to make mistakes when recovering the bit sequence that is conveyed by the transmitted signal $v(t)$.
- These so-called bit errors cause a 0 to be recovered when a 1 was transmitted and vice versa.
- The Bit Error Ratio (BER) is the fraction of bits that suffer from bit errors.
- The BER typically increases as the AWGN standard deviation σ is increased.

Channel coding (CEP 1.4.3.3)

- Channel coding can be used to allow a receiver to mitigate the bit errors in its recovered bit sequence.
- The transmitter's channel encoder inserts some carefully chosen redundancy into the bit sequence.
- A channel decoder mitigates the bit errors in the information bits by considering the redundant bits that were inserted by the channel encoder.
- The simplest examples of a channel code are a repetition code and the Hamming code. Other (more sophisticated) channel codes include BCH, convolutional, turbo, LDPC and polar codes.

Repetition coding

- For each information bit in the bit sequence, the repetition encoder inserts two redundant replicas of it - e.g. 00110 becomes 000000111111000.
- Suppose that noise in the channel causes the 1^{st}, 5^{th}, 11^{th} and 12^{th} bits in this sequence to suffer bit errors, giving 100010111100000.
- The repetition decoder considers three bits at a time and uses them to vote for the value of the decoded bit, giving 00100.
- Note that not all of the bit errors were corrected because a repetition code isn't a very strong code.
- A repetition encoder has a coding rate of $R_c = k/n = 1/3$ because it outputs $n = 3$ encoded bits whenever $k = 1$ information bit is input.
- Using channel coding and M-ary shift keying (see Lecture 6), the symbol rate is given by $R_s = R_b/(\log_2(M)R_c)$, where R_b is the information bit rate.

Channel capacity (CEP 6.6.3)

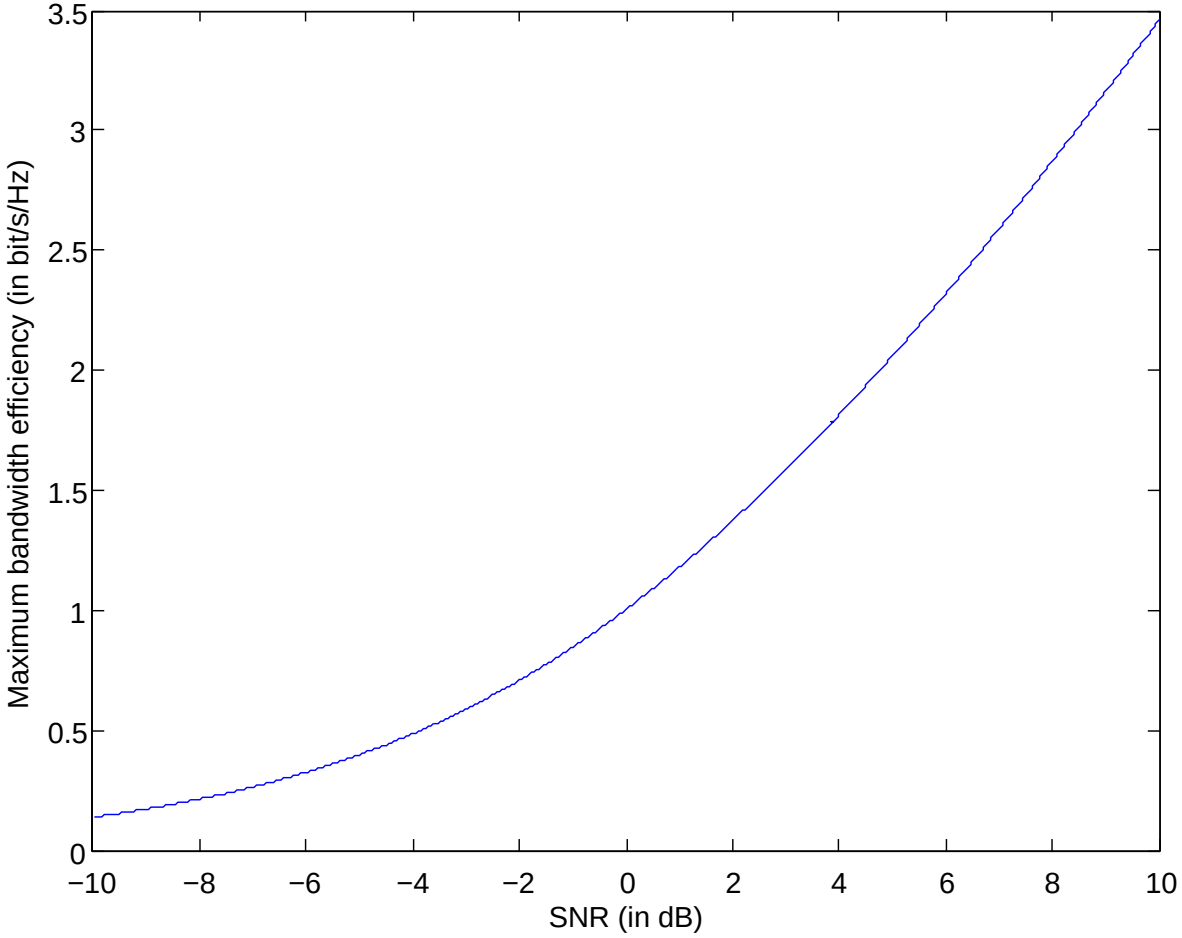
Shannon's channel capacity law states that a channel code can be designed to mitigate (nearly) all of the bit errors caused by a noisy channel, provided that the number of information bits transmitted per second R_b does not exceed the channel capacity C, i.e. if $R_b < C$.

The capacity of an AWGN channel (in bits per second) is given by $C = B \log_2(1+S/N)$ where:

- B is the bandwidth used to convey the bit sequence (in Hertz),
- $S = \lim_{T\to\infty} \frac{1}{T} \int_0^T |v(t)|^2 dt$ is the signal power (in Watts),
- $N = \lim_{T\to\infty} \frac{1}{T} \int_0^T |n(t)|^2 dt = \sigma^2$ is the noise power (in Watts),
- S/N is the Signal to Noise Ratio (SNR), which may be expressed in decibels as $10 \log_{10}(S/N)$.

Bandwidth efficiency

Bandwidth efficiency is given by $\eta = R_b/B$ in bits/s/Hz (see Lecture 7). Therefore it is possible to mitigate (nearly) all bit errors if $\eta < \log_2(1 + S/N)$.



Implication for M-ary shift keying schemes

• Using a channel coding rate R_c, the information bit rate R_b and the symbol rate R_s are related according to

$$R_b = \log_2(M) R_c R_s.$$

• For M-ary ASK, PSK and digital-QAM, the bandwidth is given by $B = R_s$ (see Lecture 7), resulting in a bandwidth efficiency of

$$\eta = \frac{R_b}{B} = \frac{\log_2(M) R_c R_s}{R_s} = \log_2(M) R_c.$$

• For M-ary FSK, the bandwidth is given by $B = \frac{1}{2}(M+1)R_s$ (see Lecture 7), resulting in a bandwidth efficiency of

$$\eta = \frac{R_b}{B} = \frac{\log_2(M) R_c R_s}{\frac{1}{2}(M+1)R_s} = \frac{2\log_2(M)}{M+1} R_c.$$

• Therefore, it is possible to mitigate (nearly) all transmission errors when using any information bit rate R_b over an AWGN channel having any SNR S/N, provided that we choose a channel coding rate R_c that reduces the bandwidth efficiency so that it satisfies $\eta < \log_2(1 + S/N)$.

• However, a lower bandwidth efficiency η implies that we require a higher bandwidth B.

Hamming code

- A Hamming code is a more sophisticated channel code than a repetition code.
- A sequence $\mathbf{x}$ of $k = 4$ information bits are input into a Hamming encoder at a time.
- In response, the Hamming encoder outputs a sequence $\mathbf{y}$ of $n = 7$ encoded bits.
- The coding rate of the Hamming code is therefore $R_c = k/n = 4/7$.

Hamming encoding

- Hamming encoding is achieved using the generator matrix $\mathbf{G}$ and the modulo-2 matrix product

$$\mathbf{y} = \mathbf{G} \cdot \mathbf{x}$$

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{G}} \cdot \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}}_{\mathbf{x}}$$

- For example

$$\begin{aligned} y_1 &= 1 \cdot x_1 \oplus 1 \cdot x_2 \oplus 0 \cdot x_3 \oplus 1 \cdot x_4 \\ &= x_1 \oplus x_2 \oplus x_4 \\ y_2 &= x_1 \oplus x_3 \oplus x_4 \\ y_3 &= x_1 \\ &\vdots \end{aligned}$$

- Here, $a \oplus b$ is the modulo-2 sum of the binary values a and b.

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

- For example, when $\mathbf{x} = [\ 1 \quad 0 \quad 1 \quad 1\ ]^T$, we get $\mathbf{y} = [\ 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1\ ]^T$.

Hamming decoding

• When the codeword $\mathbf{y}$ is transmitted over a noisy channel it will be received as $\hat{\mathbf{y}}$, which may contain some bit errors.

• Hamming decoding can mitigate the bit errors errors using the parity check matrix $\mathbf{H}$ and the modulo-2 matrix product

$$\mathbf{s} = \mathbf{H} \cdot \hat{\mathbf{y}}$$

$$\underbrace{\begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix}}_{\mathbf{s}} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}}_{\mathbf{H}} \cdot \underbrace{\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \hat{y}_4 \\ \hat{y}_5 \\ \hat{y}_6 \\ \hat{y}_7 \end{bmatrix}}_{\hat{\mathbf{y}}}$$

• Using the resultant $n - k = 3$-bit syndrome $\mathbf{s}$, the Hamming decoder decides if it thinks there are any bit errors in $\hat{\mathbf{y}}$.

Hamming decoding cont

- If the syndrome is $\mathbf{s} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$ then the Hamming decoder thinks there are no bit errors in $\hat{\mathbf{y}}$ (it may be wrong though).
- In this case, it outputs $\hat{\mathbf{x}} = \begin{bmatrix} \hat{y}_3 & \hat{y}_5 & \hat{y}_6 & \hat{y}_7 \end{bmatrix}^T$ since $y_3 = x_1$, $y_5 = x_2$, $y_6 = x_3$ and $y_7 = x_4$ in $\mathbf{G}$.
- If the syndrome $\mathbf{s}$ is not equal to $\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$ then its 3-bit number is converted into a decimal number $i \in [1, 7]$.
- In this case, the Hamming decoder thinks that the i^{th} bit in $\hat{\mathbf{y}}$ has been flipped by a bit error (it may be wrong though).
- The Hamming decoder flips the i^{th} bit in $\hat{\mathbf{y}}$ before outputting $\hat{\mathbf{x}} = \begin{bmatrix} \hat{y}_3 & \hat{y}_5 & \hat{y}_6 & \hat{y}_7 \end{bmatrix}^T$.

Some examples

$\mathbf{x}$	$\mathbf{y}$	$\hat{\mathbf{y}}$	$\mathbf{s}$	$\hat{\mathbf{x}}$	Notes
1000	1110000	1110000	000	1000	The Hamming decoder correctly determines that there are no bit errors in $\hat{\mathbf{y}}$.
1101	1010101	1000101	011	1101	The Hamming decoder correctly identifies and corrects the bit error in the 3^{rd} bit of $\hat{\mathbf{y}}$. A Hamming decoder can correct a single bit error in $\hat{\mathbf{y}}$.
0011	1000011	0001011	101	0111	The Hamming decoder correctly determines that there are bit errors in $\hat{\mathbf{y}}$, but it incorrectly thinks that the 5^{th} bit is in error. A Hamming decoder can detect the presence of two bit errors in $\hat{\mathbf{y}}$, but it cannot correct them.
0000	0000000	0010110	000	1110	The Hamming decoder incorrectly thinks that there are no bit errors in $\hat{\mathbf{y}}$. A Hamming decoder cannot (always) detect the presence of three bit errors in $\hat{\mathbf{y}}$.

Legitimate permutations

• There are $2^k = 16$ possible permutations of the Hamming encoder's $k = 4$-bit input $\mathbf{x}$.

• Although there are $2^n = 128$ possible permutations of a $n = 7$ bit sequence, only $2^k = 16$ of these are legitimate permutations of the encoder's $n = 7$-bit output $\mathbf{y}$.

• For example, 1110000 and 0000000 are legitimate permutations of $\mathbf{y}$, as shown on the previous slide.

• By contrast, 1000000 is not a legitimate permutation of $\mathbf{y}$, for example.

• Whenever $\hat{\mathbf{y}}$ is not a legitimate permutation of $\mathbf{y}$, the syndrome $\mathbf{s}$ will have a value other than 000 and the Hamming decoder can be sure that there are bit errors in $\hat{\mathbf{y}}$.

• In this case, the syndrome $\mathbf{s}$ identifies which bit of $\hat{\mathbf{y}}$ can be toggled to give the legitimate permutation of $\mathbf{y}$ that is most similar.

Hamming distances

• In general, channel decoders assume that there are a low number of bit errors in $\hat{\mathbf{y}}$ and therefore attempt to correct them by selecting the legitimate permutation of $\mathbf{y}$ that has the lowest number of differing bit values.

• The number of bits that differ between two bit sequences is called the Hamming distance d.

• For example, $d(1110000, 0000000) = 3$.

• The strength of a channel code is proportional to the minimum $d_{\min}$ of the Hamming distances between each pairing of the legitimate permutations of $\mathbf{y}$.

• For an $R_c = 4/7$ Hamming code, $d_{\min} = 3$.

• In general, a channel code can correct up to $\lfloor \frac{d_{\min}-1}{2} \rfloor$ bit errors and can detect $d_{\min} - 1$ bit errors.

Exercise

For the case of a $R_c = 4/7$ Hamming code, fill-in the empty boxes in the following table. For each case, explain if the decoder is able to detect the presence of bit errors and if it is able to correct them.

$\mathbf{x}$	$\mathbf{y}$	$\mathbf{e}$	Number of bit errors	$\hat{\mathbf{y}} = \mathbf{y} \oplus \mathbf{e}$	$\mathbf{s}$	$\hat{\mathbf{x}}$
0001		0001000				
		0000000				1001
			0	0011001		
			1	1010111		
	1101001			0100000		
	0111100		1		110	
1011				1110011		

Exercise cont

$\mathbf{x}$	$\mathbf{y}$	$\mathbf{e}$	Number of bit errors	$\hat{\mathbf{y}} = \mathbf{y} \oplus \mathbf{e}$	$\mathbf{s}$	$\hat{\mathbf{x}}$
0111			0			
	0010110			1110110		
1101		1110000				
		0000100		1011110		
1000		0010010				
1001					001	1001
0000					111	0101