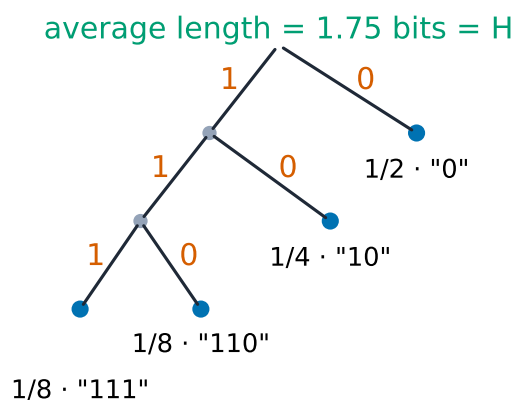
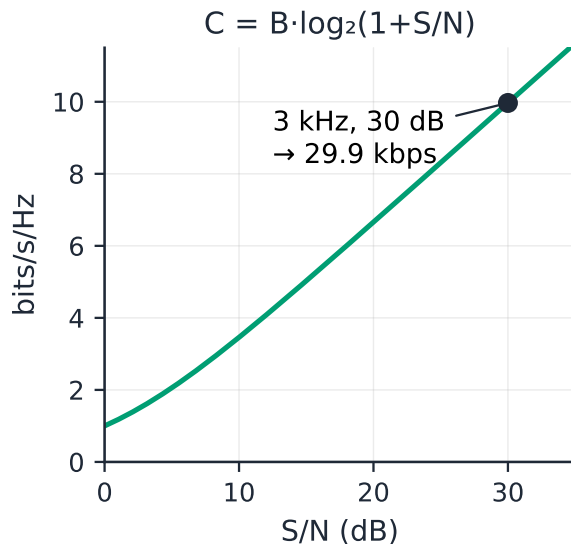


Total: 24 marks

Objective

1 Key knowledge and worked examples

$H = -\sum p \cdot \log_2 p$ bits/symbol: the quartile source (1/2, 1/4, 1/8, 1/8) gives $H = 1.75$ against 2 fixed – Huffman lengths 1,2,3,3 average exactly 1.75 ✓ (dyadic probabilities meet the bound).



Optimal lengths average exactly the entropy for dyadic probabilities; capacity is approached, never beaten.

- Capacity

$C = B \cdot \log_2(1 + S/N)$: 3 kHz at 30 dB $\rightarrow$ 29.9 kbps – the dial-up ceiling, approached (V.90 got close) but never beaten.

- Hamming (7,4)

4 data + 3 parity, distance 3: any single error is CORRECTED – encode 1011 $\rightarrow$ 0 1 1 0 0 1 1; flip bit 5 and the syndrome ($s_3s_2s_1$ read as binary) points at 5 exactly ✓.

2 Practice

2.1 Compute the entropy of a uniform 4-symbol source. [3]

2.2 Compute the average Huffman codeword length for $(0.4, 0.3, 0.2, 0.1)$ with lengths 1, 2, 3, 3. [3]

2.3 State the rate of Hamming (7,4) and how many errors it corrects. [3]

3 Exam-style questions

3.1 A source emits symbols with probabilities 0.4, 0.3, 0.2, 0.1.

Compute the entropy. Build a Huffman code (show the merging order) and compute its average length. Compute the code efficiency against entropy and the redundancy of the 2-bit fixed code. State why the Huffman average exceeds entropy slightly for this source. [8]

3.2 Hamming (7,4) protects a link; the receiver computes the syndrome from the received 7-bit word.

Give the parity-check structure (which bit positions each of s_1 , s_2 , s_3 covers). Encode the data word 0110. Flip bit 3 (counting from 1) in the codeword and compute the syndrome, showing that it identifies the error position. State what happens for a DOUBLE error and how the extended (8,4) code (distance 4) changes that. [7]

Solutions

2.1

- $H = 4 \times (1/4 \times 2) = 2\text{bits}$ (uniform = maximum uncertainty).

2.2

- $0.4 \times 1 + 0.3 \times 2 + 0.2 \times 3 + 0.1 \times 3 = 1.9\text{bits}$.

2.3

- Rate $4/7 \approx 0.571$; corrects any single error (distance 3).

3.1

- $H = 0.4 \times 1.322 + 0.3 \times 1.737 + 0.2 \times 2.322 + 0.1 \times 3.322 = 1.846\text{bits}$.
- Merge $0.1+0.2 \rightarrow 0.3$; $0.3+0.3 \rightarrow 0.6$; $0.4+0.6 \rightarrow 1.0$.
- Lengths: $0.4 \rightarrow 1$; $0.3 \rightarrow 2$; $0.2 \rightarrow 3$; $0.1 \rightarrow 3$; average 1.9.
- Efficiency $1.846/1.9 = 97.2\%$; fixed 2-bit redundancy $1 - 1.846/2 = 7.7\%$.
- The probabilities are not dyadic (no power-of-two relation), so no integer-length code can touch H exactly – Huffman is optimal over integer lengths, hence 0.054 bits/symbol above.

3.2

- s_1 covers positions $\{1,3,5,7\}$, s_2 $\{2,3,6,7\}$, s_3 $\{4,5,6,7\}$.
- $0110 \rightarrow p_1 = 0 \oplus 1 \oplus 0 = 1$, $p_2 = 0 \oplus 1 \oplus 0 = 1$, $p_3 = 1 \oplus 1 \oplus 0 = 0$: codeword $1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0$.
- Flip bit 3: $s_1 = 1$ (bit 3 $\in \{1,3,5,7\}$), $s_2 = 1$, $s_3 = 0 \rightarrow$ syndrome $s_3s_2s_1 = 011_2 = 3$ ✓ the error's position.
- A double error gives a NON-ZERO syndrome pointing at a wrong position – it mis-corrects (corrupts a third bit) silently; the extended (8,4) code adds an overall parity bit (distance 4): doubles are DETECTED (and singles still corrected) at no extra data cost beyond the eighth bit.