Solutions

Q1(a)

In communications **information** is a measurable quantity with a precise definition. If a message *m* has a probability *P(m)* then the information conveyed by the message is

$$I = \log \frac{1}{P(m)} = -\log P(m)$$

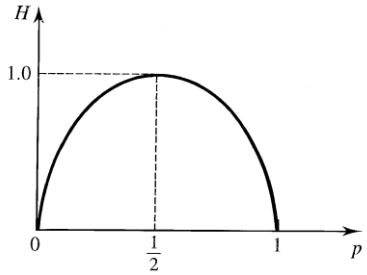
log2 is usually used to give the information in bits.
(2 marks)
The **entropy** of a source is a measure of its randomness. The more random a source the higher its entropy.
(1 mark)
Entropy is defined as

$$H = \sum_{\text{All Symbols}} -P(\text{symbol}) \log_2 (P(\text{symbol}))$$

(1 mark)
The highest entropy occurs when the symbols have equal probabilities.
(1 mark)
As an example, consider a two symbol (binary) source where *p* is the probability of transmitting a '1' we have:

$$H = p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p}$$

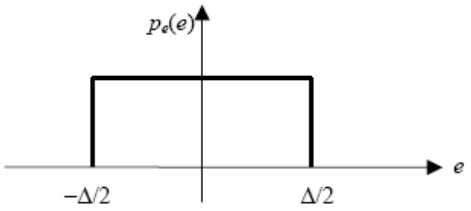
(2 marks)



The maximum entropy value 1.0 is achieved when we have p=0.5.
(1 mark)

Q1(b)

For uniform quantisation and uniformly distributed input signals, the quantisation noise variance σ is given by



$$\sigma^2 = \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} e^2 p_e de = \frac{1}{\Delta} \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} e^2 de = \frac{\Delta^2}{12}$$

Where Δ is the quantisation stepsize and p_e is the probability density function of the quantisation error.
(2 marks)
In this problem, Δ=4/64=1/16. Then the quantisation noise power P_n is given by P_n=(1/16)*(1/16)/12=0.00032552.
(2 marks)
The power P_s of the signal can be calculated in a similar was as its value is uniformly distributed over the range between -2V and 2V. We have P_s=4*4/12=4/3.
(1 mark)
Then the quantiser noise-to-signal power ratio is
$R_{ns}=P_n/P_s$=0.000244=-36dB
(1 mark)

Q1(c)

E_b/N_0 is related to the system signal to noise ratio through the following identities for binary systems (i.e. systems that send 1 bit per symbol):

$$\frac{E_b}{N_0}=\frac{ST}{N_0}=\frac{S}{RN_0}=\frac{S}{N_oB}\left(\frac{B}{R}\right)$$

where

$$S=\text{average signal power}$$
$$T=\text{symbol (bit) time period}$$
$$R=\text{bit rate (data rate)}$$

(1 mark)

For the case where transmission bit rate is equal to the channel capacity, R=C, we have

$$\frac{E_b}{N_0}=\frac{S}{N_oB}\left(\frac{B}{C}\right)$$

(1 mark)
Using

$$\frac{C}{B}=\log_2(1+\frac{S}{N_oB})$$

i.e.

$$\frac{C}{B}=\log_2\left(1+\frac{E_b}{N_o}\left(\frac{C}{B}\right)\right)$$

(1 mark)
let

$$x=\frac{E_b}{N_o}\left(\frac{C}{B}\right)$$

then

$$\frac{C}{B} = x \log_2 (1+x)^{\frac{1}{x}}$$

i.e.

$$1 = \frac{E_b}{N_o} \log_2 (1+x)^{\frac{1}{x}}$$

(1 mark)

In the limit, as $\frac{C}{B} \to 0$ and using

$$\lim_{x \to 0} (1+x)^{1/x} = e$$

(1 mark)
we get

$$\frac{E_b}{N_o} = \frac{1}{\log_2 e} = 0.693 = -1.59 dB$$

(1 mark)

Q2(a)

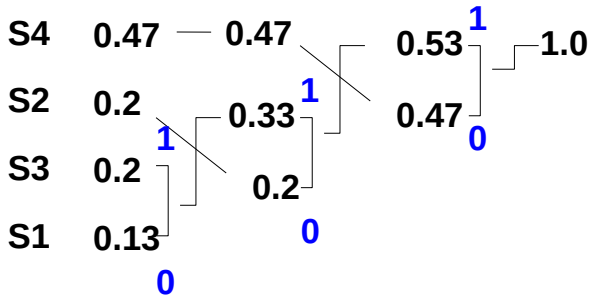
(i) Let the source be referred to as X then the Entropy is given by:
H(X) = P(0)*H(X|0) + P(1)*H(X|1).
(1 mark)

We begin by solving for the *a priori* probabilities. This can be done using the fact that:
P(1) + P(0) = 1
Substituting this into
P(0) = P(0)P(0|0) + P(1)P(0|1)
(1 mark)
and using the conditional probabilities indicated in the Markov model [P(1|1) = 0.7, etc] we get
P(0) = 1/3 and therefore P(1) =2/3 .
(1 mark)

The conditional entropies H(X|0) and H(X|1) are then computed:
H(X|0) = P(0|0)*log2(1 / P(0|0)) + P(1|0)*log2(1 / P(1|0)) = 0.9710
(1 mark)
H(X|1) = P(1|1)*log2(1 / P(1|1)) + P(0|1)*log2(1 / P(0|1)) = 0.8813
(1 mark)
So finally,
H(X) = P(0)*H(X|0) + P(1)*H(X|1) = 0.9710/3 + 0.8813*2/3
= 0.9112 bits / symbol
(1 mark)

(ii) To determine a unique, prefix-free coding scheme for this source close to its entropy, we can use extension code by using groups of two symbols as the basis and then apply Huffman coding to this new set of symbols.
(1 mark)

The new set of symbols are S1=00, S2=01, S3=10, and S4=11. Their associated probabilities are respectively:
p(00)=p(0)*p(0|0)=0.4/3=0.13
p(01)=p(0)*p(1|0)=0.6/3=0.2
p(11)=p(1)*p(1|1)=0.7*2/3=0.47
p(10)=p(1)*p(0|1)=0.3*2/3=0.2
(1 mark)
Apply Huffman coding to these four symbols we have



So, the results are:
S1: 110
S2: 10
S3: 111
S4: 0
(2 marks)
[Note: variations are possible depending on the exact algorithm used. However, to be acceptable, the code must be uniquely decodable, prefix free and efficient]

(iii) The compression ratio is the original code length divided by
the new average code length.
The average code length is given by:

$$\bar{n} = \sum_{i=1}^{4} n_i P_i$$

where n_i is the bit length of the codeword for symbol S_i, and P_i is the probability for S_i.
average length $\bar{n}$ = 0.13*3+0.2*2+0.2*3+0.47*1=1.86 bits / symbol
(1 mark)
For four symbols, a fixed length code would need to use 2 bits per symbol, therefore the compression ratio is:

$$\frac{2}{\bar{n}} = 1.0753$$

(1 mark)

Q2(b)

(i) The minimum Hamming distance between the codes is $D_{min} = 2$. Therefore, $D_{min} - 1 = 1$ bit errors can be detected per code word.
(2 marks)

(ii) Forward error correction is impossible for 1 bit errors since a 1 bit error will result in a code which has the same distance to the original codeword as that to any other valid codeword. ($D_{min}/2 - 1 = 0$)
(1 mark)
(iii) For eight messages we would need 3 bits / message. The linear block code in question uses n = 4 bits / codeword. Therefore the code rate is k/n = 3/4 = 75%.
(1 mark)

Q2(c)

Additive White Gaussian Noise (AWGN) refers to a random signal with a constant power spectral density (white) and a Gaussian distributed amplitude:
(1 mark)
The term 'additive' indicates that this random noise signal is added to the original signal as it passes through the channel.
(1 mark)
AWGN is an appropriate model for noise because:
1) Noise in real systems results from the combination of interference from many different random sources, and a result from statistics (the central limit theorem) states that the sum of a large number of random variables (possibly each with different distributions) is a random variable with a distribution that approaches a Gaussian distribution.
2) Many types of noise (notably thermal noise) have constant power spectrums over the operating bandwidths of communications systems and thus a white noise source makes a useful model.
(2 marks)

Q3(a)

When receiving digital signals it is necessary to synchronise the receiver with the incoming signal. There are three levels of synchronisation.
1. Synchronising the Carrier.
We need to lock the receiver to the incoming carrier frequency. We can use a Phase-Lock-Loop (PLL). Communication systems can operate without carrier synchronisation by using non-coherent detection, but the probability of error is then increased in most cases.
(2 marks)
2. Symbol synchronisation.
This is a higher level of synchronisation where the receiver tries to detect each individual symbol (i.e. each bit in a binary system). Bit synchronisation can be achieved using an Early/Late gate synchroniser.
(2 marks)
3. Frame synchronisation.
This is the highest level of synchronisation, where the aim is to identify blocks of data (frames) within the data stream. One approach would be to put a special voltage waveform in the signal (similar to having a full stop in a sentence) but this requires special circuitry. A better idea is to insert a code marker at the beginning of a frame and try and find this at the receiver. Barker code and Willard Code are two examples of such a code marker.
(2 marks)

Q3(b)

For sending 0, the probability of no errors is α^2 , while the probability of two errors is $(1-\alpha)(1-\beta)$; for sending 1, the probability of no errors is β^2, while the probability of two errors is $(1-\beta)(1-\alpha)$. So the overall probability of correct transmission is
(1 mark)

$$\begin{aligned} p_c &= p(1|1)\times p(1) + p(0|0)\times p(0) \\ &= p(1)\times(\beta^2 + (1-\alpha)(1-\beta)) + p(0)\times(\alpha^2 + (1-\alpha)(1-\beta)) \\ &= 0.6\times(0.64+0.02) + 0.4\times(0.81+0.02) \\ &= 0.7280 \end{aligned}$$

(2 marks)
Then the overall error probability of the cascaded channel is

$$p_e = 1 - p_c = 0.272$$

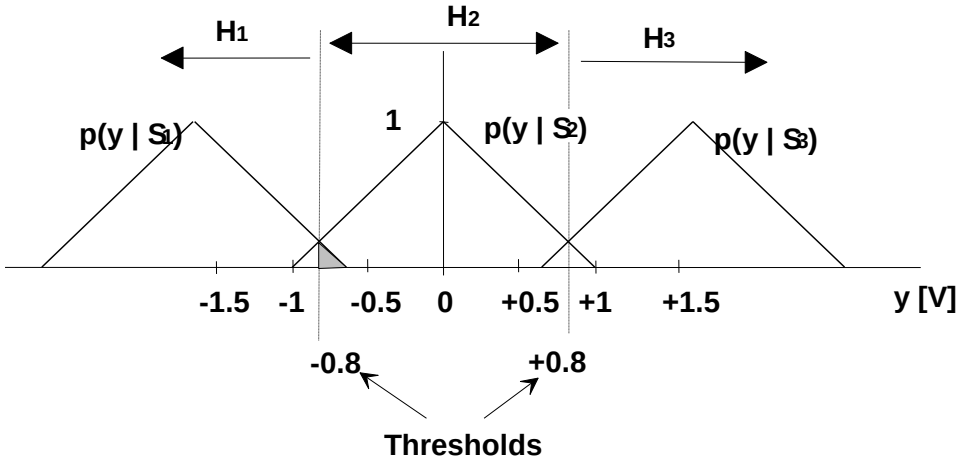
(1 mark)

Q3(c)

(i)For a maximum likelihood detector, we choose the hypothesis that has the highest likelihood value for the observed signal when compared with all possible hypotheses:

$$\hat{H}_{ML} = \arg\max_i \{p(y|S_i)\}$$

Considering a graphical representation of the situation:



It is clear that the decision thresholds are at –0.8V and +0.8V. The full decision rule is as follows:
Choose H1 if y < -0.8,
Choose H2 if –0.8< y < 0.8, and
Choose H3 if y > 0.8
(3 marks)

(ii) The probability of error is calculated by considering all the different possible ways of making an error:
Pe = P(H1|S2)P(S2) + P(H1|S3)P(S3) + P(H2|S1)P(S1) + P(H2|S3)P(S3) + P(H3|S1)P(S1) + P(H3|S2)P(S2)

From the diagram, it is clear that P(H1|S3) and P(H3|S1) are both zero.
P(H2|S1) is the shaded area under the graph of p(y|S1) so

$$P(H_2 \mid S_1) = \int_{-0.8}^{0.8} p(y \mid S_1)dy = 0.5 \times (\text{triangle base}) \times (\text{triangle height})$$

$$= 0.5 \times 0.2 \times 0.2$$

$$= 0.02$$

Also, because of symmetry, P(H1|S2) = P(H2|S3) = P(H3|S2) = P(H2|S1) so Pe = 4* P(H2|S1) * P(S1) = 0.02667
(5 marks)

(iii) If P(S2) → 1, then the decision thresholds would move outwards towards –1 and 1. In the limit the MAP error probability would tend to zero (If P(S2) = 1 then the receiver could always just choose H2 and would never make an error.)
(2 marks)

Q4(a)

Message delay is defined as: $D = w + \tau$, where
w = the average waiting time before transmission of the message begins;
τ = transmission time.
Let us assume that each user wishes to transmit messages (packets) of b bits every T seconds. We further assume that b is chosen so that the channel is fully utilised, i.e. $Mb/T = R$. Also the time slots used for the TDMA system are T/M seconds in length.

The Message delay will then be
FDMA:
$w_{fdma} = 0$ (continuous transmission)
$\tau_{fdma} = T$
so $D_{fdma} = T$
(2 marks)
TDMA:
Each packet is sent within slots of T/M seconds in length, i.e. $\tau_{tdma} = T/M$.
For the waiting time:
Packet 1 is transmitted immediately.
Packet 2 is transmitted T/M seconds later.
…
Packet m is transmitted $(m - 1)T/M$ seconds later.
The average waiting time is

$$w_{tdma} = \frac{1}{M}\sum_{m=1}^{M}(m-1)\frac{T}{M}$$

$$= \frac{T}{M^2}\sum_{m=1}^{M}(m-1)$$

$$= \frac{T}{M^2}\frac{(M-1)M}{2} = \frac{T}{2}(1-\frac{1}{M})$$

(2 marks)
So the message delay for TDMA

$$D_{tdma} = w_{tdma} + \tau_{tdma} = \frac{T}{2}(1 - \frac{1}{M}) + \frac{T}{M}$$

This is the same as

$$D_{tdma} = w_{tdma} + \tau_{tdma} = D_{fdma} - \frac{T}{2}(1 - \frac{1}{M})$$

(1 mark)

Q4(b)

(i)

Message	Parity	Codeword
000	1	1 000
100	0	0 100
010	0	0 010
110	1	1 110
001	0	0 001
101	1	1 101
011	1	1 011
111	0	0 111

(1 mark)

(ii) The code is capable of detecting all single- and triple-error patterns.
(1 mark)

(iii) The probability of an undetected error is equal to the probability that two or four errors occur anywhere in a codeword.

$$\begin{aligned} P_{nd} &= \binom{4}{2} p^2 (1-p)^2 + \binom{4}{4} p^4 \\ &= 6p^2(1-p)^2 + p^4 \\ &\approx 6 \times 10^{-6} \end{aligned}$$

(3 marks)

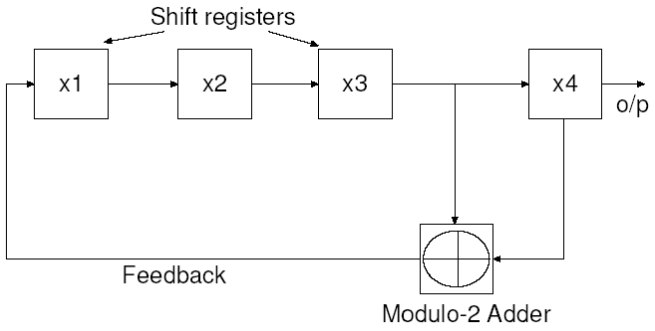
Q4(c)

(i) There are three properties:

(a) Balance property. We want roughly the same number of 0s as we have 1s in the sequence.
(b) Run property. A run sequence is a sequence of consecutive bits with the same value. We want
    a. 1/2 of run sequences to be of length 1
    b. 1/4 of run sequences to be of length 2
    c. 1/8 of run sequences to be of length 3…
(c) Correlation property. We would like the PN sequence to have very low correlation with shifted copies of itself (as with white noise). This also helps the receiver synchronise correctly.

(3 marks, 1 mark each item)

(ii) PN sequences for spread spectrum systems can be produced using circuits consisting of shift registers with feedback. An example is given below.



(2 marks)

Q4(d)

The full communication process is as follows:

1. Alice picks two prime numbers: p = 3 and q = 11 as $N = p \times q$ =33. Alice keeps those secret from everyone.
2. Alice then picks another number e = 3. With $z = (p-1) \times (q-1) = 20$. z and e are chosen in such a way that they do not have any common prime factors.
3. Alice now publishes N and e for everyone to see.
4. Before Bob sends the message M = 15 to Alice, he looks up e and N and calculates $C = M^e \bmod N = 9$. This is transmitted to Alice on a non-secure channel. Although other people can read this message, they would not be able to determine M, since a one-way function has been used to encrypt the message.
5. To decipher the message Alice uses the values only she knows, p and q, to calculate another number d so that the following equation is satisfied: $(e \times d) \bmod z = 1$, which gives a value of d = 7.
6. To decrypt the message Alice calculates $M = C^d \bmod N = 9^7 \bmod 33 = 15$.

Or you can give the simplified version:

1. Alice publishes N and e as the public key for everyone to see, and keep d secret as the private key.
2. Before Bob sends the message M = 15 to Alice, he looks up e and N and calculates $C = M^e \bmod N = 9$. This is transmitted to Alice on a non-secure channel. Although other people can read this message, they would not be able to determine M, since a one-way function has been used to encrypt the message.
3. To decipher the message Alice uses the value only she knows, i.e. d=7. To obtain the real message, Alice calculates $M = C^d \bmod N = 9^7 \bmod 33 = 15$.

(5 marks)