Indian Institute of Technology Bombay

Department of Electrical Engineering

Handout 5 EE 706 Communication Networks

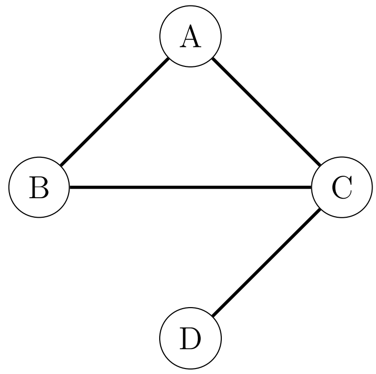
Solutions to Quiz 1 January 26, 2010

1. What is a communication network? [1 point]

   Ans. A *communication network* is a collection of communication systems and the communication links which connect them.

2. A communication network has 4 nodes A, B, C and D. A is connected to B and C, B is connected to A and C, C is connected to A, B and D. All connections are bidirectional. Draw the graph of this network. [1 point]

   Ans.



3. A source node S wants to send 100 bits of information to a destination node D.

   (a) S uses a forward error correction (FEC) scheme which adds 200 bits of redundancy to the information bit string. What is the rate of the FEC scheme? [1 point]

   Ans.

$$\text{Rate} = \frac{\text{Length of information bit string}}{\text{Length of transmitted bit string}}$$

   The length of the information bit string is 100 bits. Since the FEC scheme adds 200 bits, the length of the transmitted bit string is 300 bits. Hence we have

$$\text{Rate} = \frac{100}{100 + 200} = \frac{1}{3}$$

   (b) If an FEC scheme of rate $\frac{1}{5}$ is used by S, what is the amount of redundancy added?

   [1 point]

Ans. Let the amount of redundancy be n bits. From the formula used in part (a), we have

$$\text{Rate} = \frac{1}{5} = \frac{100}{100+n}.$$

Hence, $n = 400$ bits.

(c) If the channel between S and D has data rate equal to 10 bits per second, what is the time duration of transmission in the above two cases? [1 point]

Ans.

$$\text{Data Rate} = \frac{\text{Length of transmitted bit string}}{\text{Time taken for transmission}} = 10 \text{ bits per second}$$

$$\text{Time taken for transmission} = \frac{\text{Length of transmitted bit string}}{10} \text{seconds}$$

The length of transmitted bit string is 300 bits for part (a) and 500 bits for part (b). Hence for part (a)

$$\text{Time taken for transmission} = \frac{300}{10} = 30 \text{ seconds}$$

And for part (b)

$$\text{Time taken for transmission} = \frac{500}{10} = 50 \text{ seconds}$$

4. The 3-repetition code maps 0 to 000 and 1 to 111. It can correct one bit error. The 5-repetition code which maps 0 to 00000 and 1 to 11111 can correct 2 bit errors.

(a) How many bit errors can a 4-repetition code correct? [1 point]

Ans.

The repetition code decoder corrects bit errors by checking whether 0 or 1 are in majority, and then correcting the bits in minority to the bits in majority. For example, in the 5-repetition code if the received string is 01010, the bits in minority, i.e. 1s are corrected to the bits in majority i.e. 0s and the string becomes 00000. So the correct bits must be in majority for succesful correction of the string.

In the 4-repetition code, a 0 is mapped to 0000 and a 1 is mapped to 1111. If 2 bits are in error, then there will be an ambiguity in deciding which bits are in majority or minority. If only one bit is in error, the correct bits are in majority and the error will be corrected. Hence, a 4-repetition code can correct a maximum of 1 bit error.

(b) How many bit errors can a n-repetition code correct when n is odd? [2 points]

Ans.

Since n is odd, let $n = 2k + 1$ where k is a nonnegative integer. If k bits are in error then the remaining $k+1$ bits must be correct and hence majority decoding will correct the errors in the bit string. If more than k bits are in error then the correct bits are in minority and hence majority decoding cannot correct the errors in the bit string. So a $2k + 1$-repetition code can correct k bit errors, i.e. if n is odd, an n-repetition code can correct $\frac{n-1}{2}$ bit errors. For example, a 5-repetition code can correct $\frac{5-1}{2} = 2$ bit errors.

(c) How many bit errors can a n-repetition code correct when n is even? [2 points]

Ans.

Since n is even, let $n = 2k$ where k is a positive integer. If k bits are in error then the remaining k bits must be correct but there is an ambiguity in deciding which bits are in majority. If $k - 1$ bits are in error then the remaining $k + 1$ correct bits become in majority and hence errors in the bit string can be corrected. So, a $2k$-repetition code can correct $k - 1$ bit errors i.e. if n is even, an n-repetition code can correct $[(\frac{n}{2})$ - $1]$ bit errors. For example, a 4-repetition code can correct $(\frac{4}{2})$ - $1 = 1$ bit error.