EE 706: Communication Networks

Instructor: Saravanan Vijayakumaran
Indian Institute of Technology Bombay
Spring 2011

Solutions to Assignment 1 February 1, 2011

12. 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 400 bits of redundancy to the information bit string. What is the rate of the FEC scheme?
   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 400 bits, the length of the transmitted bit string is 500 bits. Hence we have
   $$\text{Rate} = \frac{100}{100+400} = \frac{1}{5}$$

   (b) If an FEC scheme of rate $\frac{1}{3}$ is used by S, what is the amount of redundancy added, i.e. the number of check bits added?
   Ans. Let the amount of redundancy be n bits. From the formula used in part (a), we have
   $$\text{Rate} = \frac{1}{3} = \frac{100}{100+n}.$$
   Hence, $n = 200$ bits.

   (c) If the channel between S and D has data rate equal to 20 bits per second, what is the time duration of transmission in the above two cases?
   Ans.
   $$\text{Data Rate} = \frac{\text{Length of transmitted bit string}}{\text{Time taken for transmission}} = 20 \text{ bits per second}$$
   $$\text{Time taken for transmission} = \frac{\text{Length of transmitted bit string}}{20}\text{seconds}$$
   The length of transmitted bit string is 500 bits for part (a) and 300 bits for part (b). Hence for part (a)
   $$\text{Time taken for transmission} = \frac{500}{20} = 25 \text{ seconds}$$
   And for part (b)
   $$\text{Time taken for transmission} = \frac{300}{20} = 150 \text{ seconds}$$

13. 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 6-repetition code correct? Explain your answer.

   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 6-repetition code, a 0 is mapped to 000000 and a 1 is mapped to 111111. If 3 bits are in error, then there will be an ambiguity in deciding which bits are in majority or minority. If at most two bits are in error, the correct bits are in majority and the error will be corrected. Hence, a 6-repetition code can correct a maximum of 2 bit errors.

   (b) How many bit errors can a n-repetition code correct when n is odd? Explain your answer.

   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? Explain your answer.

   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.

14. A single parity check is an error detection code which appends a single parity bit to an information bit string. The parity bit is set to 1 if the number of ones in the information bit string is odd and is set to 0 otherwise. Let the information bit string be 10101. If a single parity check bit is added to it and the resulting bit string is sent over a noisy channel, list all possible received bit strings which are declared error free at the destination. Does the set of bit strings which are declared error free at the destination depend on the transmitted bit string?

   Ans. Given that the information bit string is 10101, the parity check bit is 1 and so the transmitted bit string is 101011. The channel can introduce any number of errors and hence the received bit string can be any sequence of 6 bits. Let us denote them by $b_1b_2b_3b_4b_5b_6$. There are $2^6 = 64$ possible received bit strings. The destination

$$101011 \longrightarrow \boxed{\text{Noisy Channel}} \longrightarrow b_1b_2b_3b_4b_5b_6$$

Figure 1

recalculates the parity bit using the received information bits according the following formula.

$$\hat{b}_6 = \begin{cases} 1 & \text{Number of 1's is odd} \\ 0 & \text{otherwise} \end{cases}$$

It then checks whether the recalculated parity $\hat{b}_6$ is equal to the received parity bit b_6. If they are not equal, it declares an error. So the error free bit strings are such that $\hat{b}_6$ is equal to b_6. Since $\hat{b}_6$ is fixed once b_1, b_2, b_3, b_4, b_5 are fixed, we can only vary the latter. So there are 32 possible bit strings which are declared error-free.

$b_1b_2b_3b_4b_5b_6$	$\hat{b}_6$
000000	0
000011	1
000101	1
000110	0
001001	1
001010	0
001100	0
001111	1
010001	1
010010	0
010100	0
010111	1
011000	0
011011	1
011101	1
011110	0
100001	1
100010	0
100100	0
100111	1
101000	0
101011	1
101101	1
101110	0
110000	0
110011	1
110101	1
110110	0
111001	1
111010	0
111100	0
111111	1

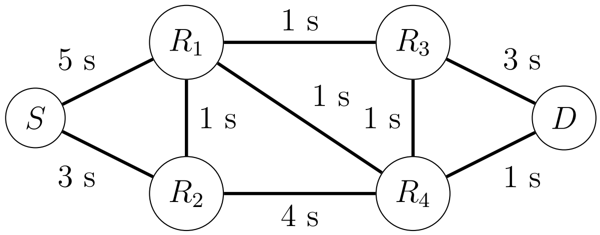
15. Consider the six-node communication network shown in the below figure.

    (a) List all routes from node S to node D.

    (b) The number alongside a link indicates the packet delay incurred on that link in seconds. Taking the routing cost of a route to be the sum of the delays of the links which constitute the route, write down the minimum-delay routing tables for the nodes S, R_1, R_2 and R_3 in the format shown in the table below.

Ans.

(a) The loop-free routes are the following



Routing table for S		
Reachable Node	Next Hop	Routing Cost

$S - R_1 - R_3 - D$
$S - R_1 - R_3 - R_4 - D$
$S - R_1 - R_4 - D$
$S - R_1 - R_4 - R_3 - D$
$S - R_2 - R_4 - D$
$S - R_2 - R_4 - R_3 - D$
$S - R_1 - R_2 - R_4 - D$
$S - R_2 - R_1 - R_3 - D$
$S - R_2 - R_1 - R_4 - D$
$S - R_2 - R_1 - R_4 - R_3 - D$
$S - R_2 - R_1 - R_3 - R_4 - D$

If routes containing loops are allowed, then there are infinitely many routes. One example is $S - R_1 - R_2 - R_4 - R_3 - R_1 - R_2 - R_4 - D$.

(b) The routing tables for S, R_1, R_2 and R_3 are given below.

Routing table for S		
Reachable Node	Next Hop	Routing Cost
R_1	R_2	4
R_2	R_2	3
R_3	R_2	5
R_4	R_2	5
D	R_2	6

Routing table for R_1		
Reachable Node	Next Hop	Routing Cost
S	R_2	4
R_2	R_2	1
R_3	R_3	1
R_4	R_4	1
D	R_4	2

Routing table for R_2		
Reachable Node	Next Hop	Routing Cost
S	S	3
R_1	R_1	1
R_4	R_1	2
R_3	R_1	2
D	R_1	3

Routing table for R_3		
Reachable Node	Next Hop	Routing Cost
S	R_1	5
R_1	R_1	1
R_2	R_1	2
R_4	R_4	1
D	R_4	2