Chun-Wei Chen (1267040)

CSE 461 HW1 – Ch 3: 3.8, 3.10, 3.11, 3.19, 3.26, 3.27, 3.28

01/28/14

3.8 To provide more reliability than a single parity bit can give, an error-detecting coding scheme uses one parity bit for checking all the odd-numbered bits and a second parity bit for all the even-numbered bits. What is the Hamming distance of this code?

This scheme can only detect single errors, a.k.a. one bit flipped, for sure. If two odd-numbered bits or two even-numbered bits flipped, the scheme will not be able to detect double errors in these kinds of scenario; however, it can detect double errors if the double errors are exactly one odd-numbered bit and even-numbered bit flipped. Making two flips to odd-numbered bits or two flips to even-numbered bits will give another valid character, so the Hamming distance is 2.

3.10 A 12-bit Hamming code whose hexadecimal value is 0xE4F arrives at a receiver. What was the original value in hexadecimal? Assume that not more than 1 bit is in error.

0xE4F -> 1 1 1 0 0 1 0 0 1 1 1
Check Bit 1 = (1 + 1 + 0 + 0 + 1 + 1) % 2 = 0
Check Bit 2 = (1 + 1 + 1 + 0 + 1 + 1) % 2 = 1
Check Bit 4 = (0 + 0 + 1 + 0 + 1) % 2 = 0
Check Bit 8 = (0 + 1 + 1 + 1 + 1) % 2 = 0
Check bit 2 is flipped, so correcting it to be 1 0 1 0 0 1 0 0 1 1 1 -> 0xA4F
The original value in hexadecimal is 0xA4F.

3.11 One way of detecting errors is to transmit data as a block of n rows of k bits per row and add parity bits to each row and each column. The bitin the lower-right corner is a parity bit that checks its row and its column. Will this scheme detect all single errors? Double errors? Triple errors? Show that this scheme cannot detect some four-bit errors.

Single errors will result in both horizontal and vertical parity checks to be incorrect. Double errors can be detected easily under this scheme. If two errors occur in the same row, the column parity will catch them. If they occur in different rows, the corresponding row parity will simply catch them. The scheme will not be able to detect all triple errors. For instance, if a data bit is flipped and the corresponding row and column parity bits are flipped as well, the scheme will not catch the error.

The scheme will simply slip the case where the bits at four corners of the n by k matrix are flipped.

3.19 In the discussion of ARQ protocol in Section 3.3.3, a scenario was outlined that resulted in the receiver accepting two copies of the same frame due to a loss of acknowledgement frame. Is it possible that a receiver may accept multiple copies of the same frame when none of the frames (message or acknowledgement) are lost?

If the latency of the transmission medium is too long so that the sender's timeout runs out before frame reached the receiver. Under this circumstance, the sender will resend the same frame again and therefore the receiver will receive two identical frames.

3.26 Suppose that the three-statement while loop near the end of protocol 6 was removed from the code. Would this affect the correctness of the protocol or just the performance? Explain your answer.

Removing the while loop will affect the correctness of the protocol. This change will lead to deadlock since that while loop is the only piece of code that handles incoming ACKs. Without this while loop, the sender will keep timing out and make no progress.

3.27 The distance from earth to a distant planet is approximately 9×10^{10} m. What is the channel utilization if a stop-and-wait protocol is used for frame transmission on a 64 Mbps point-to-point link? Assume that the frame size is 32 KB and the speed of light is 3×10^8 m/s.

Frame Size = 32 KB = 256 Kbits
Transmission Delay = Packet Size / Datarate = 256 Kb/ 64 Mbps = 0.004 s
Round-trip Time (RTT) = $2 \times (9 \times 10^{10}) / (3 \times 10^8) = 600$ s
Assume processing delay on receiver and header/ACK size is negligible since they are not mentioned in the question.
Channel Utilization = $0.004 / (0.004 + 600) = 6.667 \times 10^{-6} = 6.667 \times 10^{-4}$ %

3.28 In the previous problem, suppose a sliding window protocol is used instead. For what send window size will the link utilization be 100%? You may ignore the protocol processing times at the sender and the receiver.

RTT = 600 s
Transmission Delay = 0.004 s
In order for the utilization to be 100%, the window size should be number of packets that can be sent in one RTT. So the window size should be 600 / 0.004 = 150000