Homework 1: Link Layer

Due: 11:59 PM, Sep 28, 2018

Contents

1 Don't get me wrong

Suppose you design a code for detecting and correcting errors which works according to this table:

```
00 -> 00010
01 -> 01001
10 -> 10110
11 -> 11111
```

1. What is the rate of this code (the ratio of useful bits to total bits)?
2. What is the largest number of bit errors that this code can detect? (When we ask this, if you answer n it means the code can detect *all possilbe* n-bit errors). Justify using the Hamming-distance argument.
3. What is the largest number of bit error that this code can correct? Justify using the Hamming-distance argument.
4. Your friend suggests that you use NRZ encoding in your protocol. Why is this not a good idea for this code? Why would NRZI be much better?
5. Look at the table for 4B/5B encoding in the slides. If you were using that for error detection, what would be the largest number of chip errors you would be able to detect? (Hint: instead of computing all pairwise Hamming distances, look for examples.)

2 Flash Boys

The straight-line distance between NYC and Boston is 300Km, and the driving distance is 350km. The speed of light in vacuum (and in the air) is $c = 3 \times 10^8 m/s$, the speed of light in fiber is $2 \times 10^8 m/s$ (two thirds of c).

If you were at an HFT (high frequency trading) company in Boston you would need to minimize the time it takes to receive and send information between NYC and Boston. Assume that we have 2 options: using a radio link which goes along the direct air path (it has a bandwidth of 10Mbps, and has 6 repeaters along the path (every repeater adds $0.1ms$ to the latency of the line), or using a fiber cable which was laid along the road (it has a bandwidth of 1Gbps and no repeaters are necessary).

In the questions below, we are just worried about the one-way propagation delay, so the latency you are worried about is the time between sending the first bit, and the last bit being received by the receiver. Ignore acknowledgments, and ignore errors. Explain and show the calculations for your answers.

1. Which of the 2 links would you choose if all messages on the link are of size 100B? (It will be useful to compute the propagation, processing, and transmission delays separately.)
2. Is there any size of messages after which you would choose the other link? What size?

3 Modulation

I just looked at my Wifi card measurements: it is using a channel with a bandwidth of 80MHz, with a noise floor of -90dBm, and a received signal strength of -71dBm.

(dBm is an absolute measure of power that expresses a ratio of the measured power to a reference level of 1 miliwatt. Thus, we can directly obtain the S/N ratio (in dB) in this scenario by subtracting the two.)

1. Using Shannon's formula, what is the maximum capacity of this channel in bps?
2. Now using Hartley's law, what is maximum number of levels (assume we can only use powers of 2 for the number of levels) we can use to encode a signal?

(Out of curiosity: the newer Wifi standards, such as 802.11ac, use multiple antennas, and can have multiple spatial channels, in the same frequency, between the base station and the client. The effect is that they can multiply the achievable rate by the number of channels, which is currently up to 4.)

4 Layers

We discussed layering as a form of organizing the functionalities for communication. One challenge in designing a layered architecture is to decide where to place functionality.

Give one reason why you might want to push some functionality (say, compression, reliability, error correction, encryption, re-ordering, etc) to a lower layer, and one reason why you might want to implement some functionality at a higher layer.

Please let us know if you find any mistakes, inconsistencies, or confusing language in this or any other CS168 document by filling out the anonymous feedback form:

`https://piazza.com/brown/fall2018/csci1680`.