Calvin Passmore

ECE 6600

Homework 5

Problme 1

Two radio connections are established between four users as shown in Figure 1. User S1 transmits to user R1, and user S2 transmits to user R2. Both S1 and S2 use the same transmission power P. A minimum acceptable signal-to-interference ratio (SIR) at a receiver is 10dB. Assume that the propagation model is a power law distance dependence model, that is, $Pr = cPtd^{-\alpha}$, where Pt is the transmitted power, Pr is the received power, d is the transmitter-receiver separation distance, and α is the propagation loss exponent and c is a constant. Find the minimum distance between S1 and R (d12) such that S2 achieves acceptable SIR at R2 if the propagation loss exponent is 2 and 4, respectively. (Express d12 in terms of the distance$d$2).

Pr12 = cPtd12^-α Pr1 = cPtd1^-α

10 < log(Pr12/Pr1) = log(cPtd12^-α/cPtd1^-α) = log(d12^-α/d1^-α)

10/-α + log(d1) = log(d12)

d12 = d1 * 10^(-10/α)

Problem 2

Consider a cellular system characterized by C = 500 channels, and a minimum require signal- to-interference ratio (SIR) of 19dB in order to provide acceptable signal quality at a receiver. The cell plan used is a symmetric hexagonal plan. The system under consideration is illustrated in Figure 2. Note here that not all cells are shown in the figure. The shaded hexagons in the figure represent co-channel cells (i.e., the cells using the same set of frequencies). These cells cause so called co-channel interference to any shaded hexagon. The distance between centers of the nearest co-channel cells is denoted with D, and the radius of a cell with R. In a symmetric hexagonal cell plan, each cell has exactly 6 co-channel cells at distance D (see Figure 2). In addition, there are 6 co-channel cells at distance 3D, 6 at distance 4D, 6 at distance 7D, ..., KD, and so forth, where K = i2 + j2 +ij, j = 0; 1; 2; 3; ... (Figure 2). Recall that for the hexagonal cell plan, we have the following satisfied: Q = 3N, where N is the number of cells that divide available channels in unique and disjoint channel groups of approximately the same size. In this problem use the same propagation model as in Problem 2. In this problem use the same propagation model as in Problem 1. Also, assume that the network is of infinite size, and that all base stations transmit at the same power P.

a. Find an expression for the co-channel interference (SIR) on the downlink channel (from a base station to a terminal) that terminal A in Figure 2 experiences if the propagation loss exponent is 2 and 4, respectively (express SIR in terms of Q=D/R). Assume that all interferers at the same interference tier are at the same distance from the terminal (thus, all the first tier interferers are at distance D from the terminal). What is the problem when α = 2? Why is this not a problem in practice? NOTE: You will also need the following two in infinite sums: 1 + 1 3 + 1 4 + 1 7.... does not converge; 1 + 1 32 + 1 42 + 1 72....... = 1.233

b. The radio capacity m of a cellular system is defined as the number of users that can be supported in a single cell, that is, m = C/N [radio channels/cell] (for fixed channel allocation). What is the radio capacity m of this cellular system, assuming that α= 4?

---

a.

SIR = Pr0 / Σ_n Prn R^2 = c Pt0 d0$^{-\alpha}$ / Σ_n c Pt $d_n^{-\alpha}$ R^2 = d0$^{-\alpha}$ / $\Sigma_n d_n$ R^2

when α = 2

d0$^{-\alpha}$ / $\Sigma_n d_n$ = d0^{-2} / R^2 (3 + 4 + 7 + ...) which does not converge

when α = -4

d0$^{-\alpha}$ / $\Sigma_n d_n$ R^2 = d0^{-4} / R^2 (3^2 + 4^2 + 7^2 + ...) = d0^{-4} / R^2 (1 - 1.233) = d0^{-4} / R^2 * 0.233

There is a problem when α = 2 because the denominator doesn't converge. This isn't a problem in practive because after a certain distance, the interference isn't large enough to matter.

b.

C = 500 Q = d0^{-4} / R^2 * 0.233 N = 3*Q^2

m = C / N = C / *3Q^2 = 500 / (d0^{-4} / R^2* 0.233)^2

Problem 3

If there are 60 subscribers in a cell, 25 making a call an hour for 10 minutes each, 15 making 2 calls an hour for 6 minutes each, 10 making 3 calls an hour for 4 minutes each and 10 making 5 calls an hour for a minute each, what is the overall traffic intensity and the traffic intensity per user?

In [ ]:

```python
total_minutes_used = (25 * 10) + (15 * 2 * 6) + (10 * 3 * 4) + (10 * 5 * 1)
total_minutes_capacity = 60 * 60 # subscribers * minutes

intensity = total_minutes_used / total_minutes_capacity
intensity_user = intensity / 60

print(f'Overall Traffic intensity {intensity}')
print(f'Traffic Intensity/User {intensity_user}')
```

```
Overall Traffic intensity 0.16666666666666666
Traffic Intensity/User 0.0027777777777777775
```

Problem 4

Assume that a cellular network operator has 600 kHz of spectrum each for the uplink and downlink. With 30 kHz channels an AMPS---like FDMA, and a reuse cluster size of K = 4, determine how many users a cell can support for a 5% call---blocking rate. Assume that each user produces 35mE of load in the busy hour.

1200 kHz = total bandwidth

Channel Capacity = w/reuse * log2(1 + SIR)

SIR = Prx / (Pinter + Pn) ≈ 10

= 1,200,000 / 4 * log2(1 + 10)

```
In [ ]: from math import log2
        30000 / 4 * log2(11)
```

```
Out[ ]: 25945.73713977973
```

Problem 5

Suppose a user, Maria, discovers that her private RSA key (d1, n1) is same as the public RSA key (e2, n2) of another user, Frances. In other words, d1 = e2 and n1 = n2. Should Maria consider changing her public and private keys? Explain your answer.

Maria could consider changing her private key; however it would not be necessary. RSA encrypts and decrypts with different keys, so even if someone her private key, they don't know that it's her private key. If someone knew it was her private key then yes she should change it. The fact that her key is on the web somewhere doesn't mean that her messages are vulnerable.

Problem 6

Break the following columnar transposition cipher. The plaintext is taken from a popular computer textbook, so "computer" is a probable word. The plaintext consists entirely of letters (no spaces). The ciphertext is broken up into blocks of five characters for readability.

aauan cvlre rurnn dltme aeepb ytust iceat npmey iicgo gorch srsoc nntii imiha oofpa gsivt tpsit lbolr otoex

```
In [ ]: import math
        import numpy
        from itertools import permutations

        def cipher_solve():
            cipher = "aauancvlrerurnndltmeaeepbytusticeatnpmeyiicgogorchsrsocnntiiimihaoofpagsiv
            possible_lens = [15]

            for row_len in possible_lens:
                ## Make a table from the cipher
                cipher_table = []
                for row in range(math.ceil(len(cipher)/row_len)):
                    to_append = list(cipher[row*row_len:row*row_len+row_len])
                    cipher_table.append(to_append)
                tran_table = numpy.transpose(cipher_table)
                print(cipher_table)
                print(tran_table)

                possible_orders = permutations(range(len(tran_table[0])))
                for index_order in possible_orders:
                    output_string = ''
                    for row in tran_table:
                        for index in index_order:
```

```
                    output_string += row[index]
            print(f'{index_order}{output_string}')
            if "computer" in output_string:
                print("Found")
                return output_string
    print("Not found")

solved = cipher_solve()
print(solved)
```

```
[['a', 'a', 'u', 'a', 'n', 'c', 'v', 'l', 'r', 'e', 'r', 'u', 'r', 'n', 'n'], ['d', 'l',
't', 'm', 'e', 'a', 'e', 'e', 'p', 'b', 'y', 't', 'u', 's', 't'], ['i', 'c', 'e', 'a',
't', 'n', 'p', 'm', 'e', 'y', 'i', 'i', 'c', 'g', 'o'], ['g', 'o', 'r', 'c', 'h', 's',
'r', 's', 'o', 'c', 'n', 'n', 't', 'i', 'i'], ['i', 'm', 'i', 'h', 'a', 'o', 'o', 'f',
'p', 'a', 'g', 's', 'i', 'v', 't'], ['t', 'p', 's', 'i', 't', 'l', 'b', 'o', 'l', 'r',
'o', 't', 'o', 'e', 'x']]
[['a' 'd' 'i' 'g' 'i' 't']
 ['a' 'l' 'c' 'o' 'm' 'p']
 ['u' 't' 'e' 'r' 'i' 's']
 ['a' 'm' 'a' 'c' 'h' 'i']
 ['n' 'e' 't' 'h' 'a' 't']
 ['c' 'a' 'n' 's' 'o' 'l']
 ['v' 'e' 'p' 'r' 'o' 'b']
 ['l' 'e' 'm' 's' 'f' 'o']
 ['r' 'p' 'e' 'o' 'p' 'l']
 ['e' 'b' 'y' 'c' 'a' 'r']
 ['r' 'y' 'i' 'n' 'g' 'o']
 ['u' 't' 'i' 'n' 's' 't']
 ['r' 'u' 'c' 't' 'i' 'o']
 ['n' 's' 'g' 'i' 'v' 'e']
 ['n' 't' 'o' 'i' 't' 'x']]
(0, 1, 2, 3, 4, 5)adigitalcomputerisamachinethatcansolveproblemsforpeoplebycarryingoutin
structionsgiventoitx
Found
adigitalcomputerisamachinethatcansolveproblemsforpeoplebycarryingoutinstructionsgiventoi
tx
```

a digital computer is a machine that can solve problems for people by carrying out instructions given to it x