INDEX

6.	If a transmitter produces 50 watts of power, express the transmit power in units of (a) dBm, and (b) dBW. If 50 watts is applied to a unity gain antenna with a 900 MHz carrier frequency, find the received power in dBm at a free space distance of 100 m from the antenna, what is P (10 km)? Assume unity gain for the receiver antenna.	
7.	Determine the path loss of a 900MHz cellular system in a large city from a base station with the height of 100m and mobile station installed in a vehicle with antenna height of 2m. The distance between mobile and base station is 4Km.	
8.	Determine the path loss between base station (BS) and mobile station (MS) of a 1.8GHz PCS system operating in a high-rise urban area. The MS is located in a perpendicular street to the location of the BS. The distances of the BS and MS to the corner of the street are 20 and 30 meters, respectively. The base station height is 20m.	
9.	A mobile is located 5 km away from a base station and uses a vertical λ /4 monopole antenna with a gain of 2.55 dB to receive cellular 3 radio signals. The E-field at 1 km from the transmitter is measured to be V/rn. The carrier frequency used for this system is 900 MHz. (a) Find the length and the gain of the receiving antenna. (b) Find the received power at the mobile using the 2-ray ground reflection model assuming the height of the transmitting antenna is 50 m and the receiving antenna is 1.5m above ground.	
10.	A hexagonal cell within a 4-cell system has a radius of 1.387 km. A total of 60 channels are used within the entire system. If the load per user is 0.029 Erlangs, and λ= call/hour, compute the following for an Erlang C system that has a 5% probability of a delayed call: (a) How many users per square kilometer will this system support? (b) What is the probability that a delayed call will have to wait for more than 10s? (c) What is the probability that a call will be delayed for more than 10 seconds?	

Problem-1: If a total of 33 MHz of bandwidth is allocated to a particular FDD cellular telephone system which uses two 25 kHz simplex channels to provide full duplex voice and control channels, compute the number of channels available per cell if a system uses-

(a) 4-cell reuse. (b) 7-cell reuse. (c) 12-cell reuse.

If 1 MHz of the allocated spectrum is dedicated to control channels, determine an equitable distribution of control channels and voice channels in each cell for each of the three systems.

<u>Solution:</u>

Here given,

Total bandwidth = 33 MHz = = 33,000 kHz

Channel bandwidth = 25 kHz × 2 simplex channels= 50 kHz / duplex channel

Total available channels = Total Bandwidth / Channel Bandwidth

= 33,000 / 50

= 660 channels

If 1 MHz of the allocated spectrum is dedicated to control channels,

i.e. Control channel bandwidth = 1000 kHz

The number of available control channel = Control channel bandwidth / Channel bandwidth

= 1000 / 50

= 20 channels

(a).

For Cluster size, N=4,

Total number of channels available per cell = Total available channels / N

= 660/4 ≈165 channels.

∴ Equitable distribution of,

Voice Channel = (Total available channels - The number of available control channel) / N

= (660 – 20) / 4 ≈ 160 channels

Control Channel = Total number of channels available per cell – Voice Channel

= 165 – 160 = 5 channels

(b).

Have given,

Cluster size, N = 7

Total number of channels available per cell
= Total available channels / N
= 660/7 ≈ 94 channels.

∴ Equitable distribution of,
Voice Channel =
(Total available channels - The number of Available control channel)/N
= (660 – 20) / 7 ≈ 91 channels

Control Channel

= Total number of channels available per cell - Voice channel
= 94 – 91 = 3 channels

(c).

Here given,

Cluster size, N = 12

Total number of channels available per cell
= Total available channels / N
= 660/12 ≈ 55 channels.

∴ Equitable distribution of,

Voice Channel =
(Total available channels - The number of available control channel) / N
= (660 – 20) / 12 ≈ 53 channels

Control Channel

= Total number of channels available per cell – Voice Channel
= 55 – 53 = 2 channels

Source Code (Python):

```
import math
bw = 33000
schannel_bw = 25

print('Channel Bandwidth..')
dup_ch_bw = 2 * schannel_bw
t_ch = bw / dup_ch_bw
print(f'Duplux Channel Bandwidth: {dup_ch_bw}')
print('Total available channel:')
print(t_ch)

cc_bw = 1000
t_cc = cc_bw / dup_ch_bw
print('Total control channel:')
print(t_cc)
```

```
N = [4, 7, 12]

for i in range(3):

    ch = t_ch / N[i]
    ch_per_cell = round(ch)
    print(f'Channel per cell for N = {N[i]}:')
    print(ch_per_cell

    c = t_cc / N[i]
    cc = round(c)

    v = (t_ch - t_cc) / N[i]
    vc = round(v)

    print(f'Control channel: {cc}, Voice channel: {vc}')
```

Output:

```
Channel Bandwidth..

Duplux Channel Bandwidth: 50

Total available channel:

660.0

Total control channel:

20.0

Channel per cell for N = 4:

165

Control channel: 5, Voice channel: 160

Channel per cell for N = 7:

94

Control channel: 3, Voice channel: 91

Channel per cell for N = 12:

55

Control channel: 2, Voice channel: 53
```

Problem-2: If a signal to interference ratio of 15 dB is required for satisfactory forward channel performance of a cellular system, what is the frequency reuse factor and cluster size that should be used for maximum capacity if the path loss exponent is-

(a) $n = 4$. (b) $n = 3$.

Assume that there are 6 co-channels cells in the first tier and all of them are at the samedistance from the mobile. Use suitable approximations.

Solution:

Here given,

Minimum Required Signal-to-Noise interference ratio, S/I = 15 dB,

The number of Co-channel interfering cells, $i_o = 6$

We Know,

Number of cell reuse, $N = i^2 + i*j + j^2$………………………(i)

Also,

The Frequency Reuse Factor, Q = D/R

$= \sqrt{(3N)}$..(ii)

Where,

D = Distance between centers of the nearest Co-channel cells.

R = Radius of the cell.

Table -1: Co-channel Reuse Ratio for Some Values of N

Value of i and j	Cluster Size(N)	Co-Channel Reuse Ratio(Q)
i= 1, j = 1	3	3
i= 1, j = 2	7	4.58
i= 2, j = 2	12	6
i= 1, j = 3	13	6.24

Now,

Signal-to-Noise interference ratio, $S/I = 10 \log (Q^n / i_0)$…………………..(iii)

(a).

Have given,

Path Loss exponent, $n= 4$

Frequency Reuse Factor, Q = 4.583.

We know,

Signal-to-Noise interference ratio, S/I $= 10 \log (Q^n / i0)$

$= 10 \log ((4.583)^4 / 6)$

$= 18.66$ dB.

Since this is greater than the minimum required S/I (18.66 > 15), N = 7 can be used.

(b).

Here given,

Path Loss exponent, n = 3

We know,

Signal-to-Noise interference ratio, S/I $= 10 \log(Q^n / i_0)$ (iii)

$= 10 \log ((4.583)^3 / 6)$

$= 12.05$ dB.

Since this is less than the minimum required S/I (12.05 < 15), we need to use a larger N.

Using equation (i), the next possible value of N = 12;

For i = j = 2. The corresponding co-channel ratio is given by equation (ii) as- Frequency Reuse Factor, Q=6

Using equation (iii) the signal-to-interference ratio, S/I = 15.56 dB.

Since, this is greater than the minimum required S/I (15.56 > 15), N = 12 can be used.

<u>Source Code (Python):</u>

```python
import numpy as np

R_SI = 15
io = 6
n = [4, 3]

for a in range(2):
    N = 7
    Q = np.sqrt(3 * N)
    print('n:', n[a])
    print('Frequency reuse factor:', Q)
    SI = 10 * (np.log10((1 / io) * (Q **n[a])))

    print('Signal to interference ratio:', SI)
```

```python

if SI < R_SI:
        i, j = 2, 2
        N = (i ** 2) + (i * j) + (j ** 2)
        Q = np.sqrt(3 * N)

        print('n:', n[a])

        print('Frequency reuse factor:', Q)
        SI1 = 10 * (np.log10((1 / io) * (Q **n[a])))

        print('Signal to interference ratio:', SI1)
```

Output:

```
n: 4
Frequency reuse factor: 4.58257569495584
Signal to interference ratio: 18.66287339084195

n: 3
Frequency reuse factor: 4.58257569495584
Signal to interference ratio: 12.051776917172353

n: 3
Frequency reuse factor: 6.0
Signal to interference ratio: 15.563025007672874
```

Problem-3: How many users can be supported for 0.5% blocking probability for the following number of trunked channels in a blocked calls cleared system?

(a) 1, (b) 5, (c) 10, (d) 20, (e) 100.

Assume each user generates 0.1 Erlangs of traffic.

Solution:

Hare given,

Blocking Probability, P_B = 0.5%,

Traffic Intensity, Au = 0.1 Erlangs

We Know,

For Erlangs B, Grade of Service, GOS = P_B= 0.005

And, Total number of users, $U= A / A_u$ ……… (i)

Where, A = Offered Traffic Intensity.

Table-1: Capacity of an Erlang B System

Number of Channel C	Capacity (Erlangs) for GOS			
	= 0.01	= 0.005	= 0.002	= 0.001
2	0.153	0.105	0.065	0.046
4	0.869	0.701	0.535	0.439
5	1.36	1.13	0.900	0.762
10	4.46	3.96	3.43	3.09
20	12.0	11.1	10.1	9.41
24	15.3	14.2	13.0	12.2
40	29.0	27.3	25.7	24.5
70	56.1	53.7	51.0	49.2
100	84.1	80.9	77.4	75.2

From Table , we can find the total capacity in Erlangs for the 0.5% GOS for different numbers of channels.

(a)

Given,

C = 1 ,

A_u = 0.1,

GOS = 0.005

From Table-1, we obtain A = 0005.

Therefore, total number of users, U = A/A_u = 0.005/0.1 = 0.05 users.

But, actually one user could be supported on one channel. So, U = 1.

(b)

Given,

C = 5,

A_u = 0.1,

GOS = 0.005

From Table-1, we obtain A = 1.13.

Therefore, total number of users, U = A/A_u = 1.13/0.1 $\approx$ 11 users.

(c)

Given,

C = 10,

A_u = 0.1,

GOS = 0.005

From Table-1, we obtain A = 3.96.

Therefore, total number of users, U = A/A_u = 3.96/0.1 $\approx$ 39 users.

(d)

Given ,

C = 20,

A_u = 0.1,

GOS = 0.005

From Table-1, we obtain A = 11.10.

Therefore, total number of users, U = A/A_u = 11.1/0.1 $\approx$ 110 users.

(e)

Given

C = 100,

A_u = 0.1 ,

GOS = 0.005,

From Table-1, we obtain A = 80.9.

Therefore, total number of users, U = A/A_u = 80.9/0.1 = 809 users.

Source Code (Python):

```
import numpy as np

# Constants
Gos = 0.5 / 100  # Blocking probability
Au = 0.1         # Traffic intensity per user
A = np.array([0.005, 1.13, 3.96, 11.1, 80.9])  # Traffic intensity per configuration
c = np.array([1, 5, 10, 20, 100])              # Number of channels per configuration

# Display the values
print("Blocking probability:", Gos)
print("Traffic intensity per user:", Au)
print("Traffic intensity:", A)
print("Channels:", c)

# Calculate the number of users
U = A / Au
# Use floor rounding, except when the value is less than 1, to ensure at least 1 user
u = np.where(U < 1, 1, np.floor(U))

print("Number of users:", u.astype(int))
```

Output:

```
Blocking probability: 0.005

Traffic intensity per user: 0.1

Traffic intensity: [5.00e-03 1.13e+00 3.96e+00 1.11e+01 8.09e+01]

Channels: [  1   5  10  20 100]

Number of users: [  1  11  39 110 809]
```

Problem-4: An urban area has a population of 2 million residents. Three competing trunked mobile networks (systems A, B, and C) provide cellular service in this area. SystemA has 394 cells with 19 channels each, system B has 98 cells with 57 channels each, and system C has 49 cells, each with 100 channels. Find the number of users that can besupported at 2% blocking if each user averages 2 calls per hour at an average call duration of 3 minutes. Assuming that all three trunked systems are operated at maximum capacity, compute the percentage market penetration of each cellular provider.

Solution:

Here given,

Blocking Probability, $P_B = 2\%$,

The average number of call requests per unit time λ = 2.

The average duration of a call, H = 3/60 seconds

There are 2 million residents in the given urban area = 2000000

We Know,

For Erlangs B, Grade of Service, GOS = P_B = 0.02

And, Traffic Intensity, $A_u = \lambda H = 0.1$ Erlangs

Total number of user, $U = A / A_u$ ………… (i)

Where,

A = Offered Traffic Intensity.

Also,

Table-1: Capacity of an Erlang B System

Number of Channel C	**Capacity (Erlangs) for GOS**			
	= 0.01	**= 0.005**	**= 0.002**	**= 0.001**
2	0.153	0.105	0.065	0.046
4	0.869	0.701	0.535	0.439
5	1.36	1.13	0.900	0.762
10	4.46	3.96	3.43	3.09
20	12.0	11.1	10.1	9.41
24	15.3	14.2	13.0	12.2
40	29.0	27.3	25.7	24.5
70	56.1	53.7	51.0	49.2
100	84.1	80.9	77.4	75.2

From Table 1, we can find the total capacity in Erlangs for the 2% GOS for different number of channels.

For System-A:

Here given,

Number of channels per cell used in the system,C = 19

From table 1, For C = 19 and GOS = 0.02 we obtain

A= 12 ErlangsFrom equation (i), we have-

Total number of users, $U = A / A_u$

$= 120$ users.

Since there are 394 cells, the total number of' subscribers that can be supported by System A is equal to 120 × 394 = 47280.

Since, the percentage market penetration = 47280/2000000 = 2.36%

For System-B:

Here given,

Number of channels per cell used in the system,C = 57

From table 1, For C = 57 and GOS = 0.02 we obtain,

A = 45 Erlangs

From equation (i), we have-

Total number of user, $U = A / A_u$

$= 450$ users.

Since there are 98 cells, the total number of' subscribers that can be supported by System B is equal to 450 × 98 = 44,100.

Since, the percentage market penetration = 44100/2000000 = 2.205%

For System-C:

Here given,

Number of channels per cell used in the system,C = 100

From table 1, For C = 100 and GOS = 0.02 we obtain,

A = 88 Erlangs

From equation (i), we have-

Total number of user, $U = A / A_u$

$= 880$ users.

Since there are 49 cells, the total number of' subscribers that can be supported by System C is equal to 880 × 49 = 43,120.

Since, the percentage market penetration = 43,120/2000000 = 2.156%

Therefore, total number of cellular subscribers that can be supported by these three systemsare (47280 + 44100 + 43120) = 134500 users.

The market penetration of the three systems combined is equal to 134500/2000000 = 6.725%

Source Code (Python):

```
# Constants
blocking_p = 2 / 100  # Blocking probability
lamda = 2             # Arrival rate
H = 3 / 60            # Average call holding time in hours
Au = lamda * H        # Traffic intensity per user

# System A
print("For System A")
channel_a = 19
cell_A = 394
A = 12                # Total traffic in Erlangs
# Calculate number of users in System A
Ua = A / Au
print("Number of users in System A:", Ua)
# Calculate total number of subscribers in System A
subscriber_A = Ua * cell_A
print("Total number of subscribers in System A:", subscriber_A)
# Market penetration for System A
percentage_market_penetration_for_A = (subscriber_A / 2000000) * 100
print("Percentage market penetration for System A:",
percentage_market_penetration_for_A)

# System B
print("\nFor System B")
channel_b = 57
cell_B = 98
Ab = 45               # Total traffic in Erlangs
# Calculate number of users in System B
Ub = Ab / Au
print("Number of users in System B:", Ub)
# Calculate total number of subscribers in System B
subscriber_B = Ub * cell_B
print("Total number of subscribers in System B:", subscriber_B)
# Market penetration for System B
percentage_market_penetration_for_B = (subscriber_B / 2000000) * 100
print("Percentage market penetration for System B:",
percentage_market_penetration_for_B)

# System C
```

```
print("\nFor System C")
channel_c = 100
cell_C = 49
Ac = 88             # Total traffic in Erlangs
# Calculate number of users in System C
Uc = Ac / Au
print("Number of users in System C:", Uc)
# Calculate total number of subscribers in System C
subscriber_C = Uc * cell_C
print("Total number of subscribers in System C:", subscriber_C)
# Market penetration for System C
percentage_market_penetration_for_C = (subscriber_C / 2000000) * 100
print("Percentage market penetration for System C:",
percentage_market_penetration_for_C)
# Total subscribers and market penetration across all systems
Total_number_of_subscriber = subscriber_A + subscriber_B + subscriber_C
Market_penetration_for_three_system = (Total_number_of_subscriber / 2000000) *
100
print("\nTotal number of subscribers across all systems:",
Total_number_of_subscriber)
print("Market penetration for all three systems:",
Market_penetration_for_three_system)
```

Output:

```
For System A

Number of users in System A: 120.0
Total number of subscribers in System A: 47280.0
Percentage market penetration for System A: 2.3640000000000003

For System B

Number of users in System B: 450.0
Total number of subscribers in System B: 44100.0
Percentage market penetration for System B: 2.205

For System C

Number of users in System C: 880.0
Total number of subscribers in System C: 43120.0
Percentage market penetration for System C: 2.156


Total number of subscribers across all systems: 134500.0

Market penetration for all three systems: 6.7250000000000005
```

Problem-5: A certain city has an area of 1,300 square miles and is covered by a cellular system using a 7-cell reuse pattern. Each cell has a radius of 4 miles and the city is allocated 40 MHz of spectrum with a full duplex channel bandwidth of 60 kHz. Assume a GOS of 2% for an Erlang B system is specified. If the offered traffic per user is 0.03 Erlangs, compute-

a) The number of cells in the service area,
b) The number of channels per cell,
c) Traffic intensity of each cell,
d) The maximum carried traffic,
e) The total number of users that can be served for 2% GOS,
f) The number of mobiles per channel, and
g) The theoretical maximum number of users that could be served at one time by thesystem.

Solution:

(a)

Here given,

Total coverage area = 1300 miles Cell radius = 4 miles

We know,

The area of a cell (hexagon) can be shown to be 2.5981R2 Thus each cell covers 2.5981 × (4)2 = 41.57 sq km.

Hence, the total number of cells, Nc = 1300/41.57 = 31 cells

(b)

Here given,

Allocated spectrum = 40, 000,000 Hz
Channel width = 60,000 Hz
Frequency reuse factor, N = 7 cells

We know,

The total number of channels per cell,
C = Allocated spectrum /(Channel width × N)
= 40, 000,000 / (60,000 × 7)
= 95 channels/cell

(c)

Here given,

From (b) No, C = 95 And, GOS = 0.02
From the table 1 (Erlang B chart)
For C = 95 and GOS = 0.02,

We have,

Traffic intensity per cell, A = 84 Erlangs/cell

(d)

Here given,

From (a), Number of cells = 31 cells
From (c), Traffic intensity per cell = 84 Erlangs/cell

We Know,

Maximum carried traffic = Number of cells × Traffic intensity per cell
= 31 × 84
= 2604 Erlangs.

(e)
Here given,
Traffic per user, A_u = 0.03 Earlangs
From (d), Total traffic, A = 2604 Erlangs.
We Know,
Total number of users, U = A / Au
= 2604 / 0.03
= 86,800 users.

(f)
Here given,
Allocated spectrum = 40, 000,000 Hz
Channel width = 60,000 Hz
From (e), Number of users, U = 86,800 users.
We Know,
Number of channels = Allocated Spectrum / Channel Width
= 40, 000,000/60,000
≈ 666
Number of mobiles per channel = Number of users/Number of channels
= 86,800 / 666
≈ 130 mobiles/channel

(g)
Here given,
From (b) No, C = 95 channels/cell
From (a), the total number of cells, N_c= 31 cells.
From (e) Total number of users, U = 86,800 users.
We Know,
The theoretical maximum number of served mobiles is the number of available channels in the system (all channels occupied)
= $C \times N_c$ = 95 × 31 = 2945 users,

Which is (2945/86,800) × 100 = 3.4% of the customer base.

Source Code (Python Code):

```
import math

# Constants
area = 1300                    # Total area
radius = 4                     # Radius of each cell
allocated_spectrum = 40000     # Total allocated spectrum in Hz
channel_width = 60             # Width of each channel in Hz
frequency_reuse_factor = 7     # Frequency reuse factor
traffic_intensity_per_cell = 84  # Traffic intensity per cell
traffic_per_user = 0.03        # Traffic per user in Erlangs

# (a) Calculate number of cells that can cover the area
each_cell_covers = math.floor(2.5981 * radius**2)
number_of_cells = math.floor(area / each_cell_covers)
print("(a) Number of cells:", number_of_cells)
```

```
# (b) Calculate number of channels per cell
number_of_channel_per_cell = math.floor(allocated_spectrum / (channel_width *
frequency_reuse_factor))
print("(b) Number of channels per cell:", number_of_channel_per_cell)

# (c) Display traffic intensity per cell
print("(c) Traffic intensity per cell:", traffic_intensity_per_cell)

# (d) Calculate maximum carried traffic
maximum_carried_traffic = number_of_cells * traffic_intensity_per_cell
print("(d) Maximum carried traffic:", maximum_carried_traffic)

# (e) Calculate total number of users
total_number_of_user = maximum_carried_traffic / traffic_per_user
print("(e) Total number of users:", total_number_of_user)

# (f) Calculate number of mobiles per channel
number_of_channels = number_of_channel_per_cell * frequency_reuse_factor
number_of_mobile_per_channel = math.floor(total_number_of_user /
number_of_channels)
print("(f) Number of mobiles per channel:", number_of_mobile_per_channel)

# (g) Theoretical maximum number of users that could be served
theoretical_maximum_number_of_users = number_of_cells *
number_of_channel_per_cell
print("(g) Theoretical maximum number of users that could be served:",
theoretical_maximum_number_of_users)
```

Output:

```
(a) Number of cells: 31

(b) Number of channels per cell: 95

(c) Traffic intensity per cell: 84

(d) Maximum carried traffic: 2604

(e) Total number of users: 86800.0

(f) Number of mobiles per channel: 130

(g) Theoretical maximum number of users that could be served: 2945
```

<u>Problem No:</u> 06

<u>Name of the Problem:</u> If a transmitter produces 50 watts of power, express the transmit power in units of (a) dBm, and (b) dBW. If 50 watts is applied to a unity gain antenna with a 900 MHz carrier frequency, find the received power in dBm at a free space distance of 100 m from the antenna, what is P_r(10 km)? Assume unity gain for the receiver antenna.

<u>Solution:</u>

Given:

Transmitter power, P_t = 50 W
Carrier frequency, f_c = 900 MHz
(a) Transmitter power, $P_t(dBm) = 10\log[P_t(mW)/(1mW)]$
$= 10\log[50\times 10^3]$
$= 47.0$ dBm.
(b) Transmitter power, $P_t(dBW) = 10\log[P_t(W)/(1W)]$
$= 10\log[50]$
$= 17.0$ dBW.
The received power can be determined using equation,

$$P_r = \frac{P_t G_t\, G_r\, \lambda^2}{(4\pi)^2 d^2 L} = \frac{50(1)(1)(1/3)^2}{(4\pi)^2(100)^2(1)} = (3.5 \times 10^{-6})\text{ W} = (3.5 \times 10^{-3})\text{ mW}$$

$$P_r(\text{dBm}) = 10\log P_r(\text{mW}) = 10\log(3.5 \times 10^{-3}) = -24.5\text{ dBm}$$

The received power at 10 km can be expressed in terms of dBm using equation (1), where d_0 = 100 m and d = 10km .

$$P_r(10\text{km}) = P_r(100) + 20\log\left[\frac{100}{10000}\right] = -24.5\text{ dBm} - 40\text{ dBm}$$
$$= -64.5\text{ dBm}$$

<u>Source Code (Python Code):</u>

```
import math

Pt = 50  # Transmitter Power
fc = 900  # Carrier Frequency

# a
PtdBm = 10 * math.log10(Pt * 1e3)
print("(a)Transmitted Power: %.1f dBm" % PtdBm)

# b
PtBW = 10 * math.log10(Pt)
print("(b)Transmitted Power: %.1f dBW" % PtBW)
```

```
# received power
Gt = 1
Gr = 1
lam = 1 / 3
d = 100
L = 1
Pr = (Pt * Gt * Gr * (lam**2)) / (((4 * math.pi) ** 2) * (d**2) * L)
PrdBm = 10 * math.log10(Pr * 1e3)
print("(c)Received Power: %.1f dBm" % PrdBm)

# Pr(10Km)
Pr10Km = PrdBm + 20 * math.log10(100 / 10000)
print("(d)Received power at 10km: %.1f dBm" % Pr10Km)
```

Output:

```
(a)Transmitted Power: 47.0 dBm

(b)Transmitted Power: 17.0 dBW

(c)Received Power: -24.5 dBm

(d)Received power at 10km: -64.5 dBm
```

Problem No: **07**

Name of the Problem: Determine the path loss of a 900 MHz cellular system in a large city from a base station with the height of 100m and mobile station installed in a vehicle with antenna height of 2m. The distance between mobile and base station is 4Km.

Solution:

Have given,

The frequency, fc = 900 MHz (150 MHz to 1500MHz)

The effective transmitter (base station) antenna height, h_{te} = 100m

The effective transmitter (mobile) antenna height, h_{re} = 2m

T-R separation distance, d = 4 km

Now, The correction factor for effective mobile antenna height,

$a(h_{re}) = 3.2 (\log 11.75\, hre)^2 - 4.97$ dB for fc $\geq$ 300 MHz

$= 1.045$ dB

From Okumura-Hata Model we know, The path loss in urban areas is given by,

$$L_p = 69.55 + 26.16 \log f_c - 13.82 \log h_{te} - a(h_{re}) + [44.9 - 6.55 \log h_{te}] \log d$$

$= 137.3$ dB

Source Code (Python Code):

```
import math

# Given values
hre = 2  # Height of the receiving antenna (meters)
hte = 100  # Height of the transmitting antenna (meters)
fc = 900  # Frequency (MHz)
d = 4  # Distance between antennas (kilometers)

# Calculate a_hre
a_hre = 3.2 * (math.log10(11.75 * hre)) ** 2 - 4.97

# Calculate path loss
Lp = 69.55 + 26.16 * math.log10(fc) - 13.82 * math.log10(hte) - a_hre + (44.9 - 6.55 *
math.log10(hte)) * math.log10(d)

print('The path loss in urban areas: %.2f' % Lp)
```

Output:

The path loss in urban areas: 137.29

Problem No: **08**

Name of the Problem: Determine the path loss between base station (BS) and mobile station (MS) of a 1.8GHz PCS system operating in a high-rise urban area. The MS is located in a perpendicular street to the location of the BS. The distances of the BS and MS to the corner of the street are 20 and 30 meters, respectively. The base station height is 20m.

Solution:

Have given,

The frequency, fc = 1.8 GHz (0.9 to 2 GHz)

The effective transmitter (base station) antenna height, hb = 20m

From Okumura-Hata Model we know,The path loss in a high-rise urban areas with Perpendicular Street to the location of the Base Station is given by The distance of the mobile from the base station is, d = $\sqrt{20^2 + 30^2}$ = 36.05m.

Using the appropriate equation, we can write the path loss as:

$$L_p = 135.41 + 12.49logf_c - 4.99logh_b + [46.84 - 2.34logh_b]logd$$

$$= 68.89 \text{ dB}$$

Source Code (Python Code):

```
import math
# Given values
fc = 1.8  # Frequency (GHz)
hb = 20  # Height of the base station antenna (meters)
d = math.sqrt(20 ** 2 + 30 ** 2) / 1000  # Distance between the base station and mobile station (kilometers)
print('Distance: %.4f Km' % d)

# Calculate path loss
PathLoss = 135.41 + (12.49 * math.log10(fc)) - (4.99 * math.log10(hb)) + ((46.82 - 2.34 * math.log10(hb)) * math.log10(d))

print('PathLoss: %.4f dB' % PathLoss)
```

Output:

Distance: 0.0361 Km

Path Loss: 68.9368 dB

<u>Problem No:</u> 09

<u>Name of the Problem:</u> A mobile is located 5 km away from a base station and uses a vertical λ /4 monopole antenna with a gain of 2.55 dB to receive cellular 3 radio signals. The E-field at 1 km from the transmitter is measured to be 10^{-3} V/m. The carrier frequency used for this system is 900 MHz. (a) Find the length and the gain of the receiving antenna. (b) Find the received power at the mobile using the 2-ray ground reflection model assuming the height of the transmitting antenna is 50 m and the receiving antenna is 1.5m above ground.

<u>Solution:</u>

Given,

T-R separation distance = 5 km

E-field at a distance of 1 km = 10^{-3} V/m

Frequency of operation, f = 900 MHz

$$\lambda = \frac{c}{f} \frac{3\times10^{8}}{900\times10^{6}} = 0.333 \text{ m}$$

(a) Length of the antenna, L = λ /4 = 0.333/4 = 0.0833 m = 8.33 cm.

Gain of λ /4 monopole antenna can be obtained using equation.

Gain of antenna =1.8 = 2.55 dB.

(b) Since d ≫ $\sqrt{h_t h_r}$, the electric field is given by,

$$E_R(d) = \frac{2E_0 d_0}{d} \frac{2\pi h_t h_r}{\lambda \text{d}} = \frac{k}{d^2} \text{ V/m}$$

$$= \frac{2\times10^{-3}\times1\times10^{3}}{5\times10^{3}} [\frac{2\pi(50)(1.5)}{0.333(5\times10^{3})}]$$

$$= 113.1 \times 10^{-6} \text{ V/m}$$

The received power at a distanced can be obtained using equation (2),

$$P_r(d) = \frac{(113.1\times10^{-6})^2}{377} [\frac{1.8(0.333)^2}{4\pi}]$$

$$P_r(d = 5\ km) = 5.4 \times 10^{-13}\ W$$

$$= -122.68 \text{ dBW or } -92.68 \text{ dBm.}$$

Source Code (Python Code):

```
import math
# Constants
f = 900  # Frequency in MHz
g = 2.55  # Gain of antenna in dB

# Question (a)
gain = 10 ** (g / 10)  # Linear gain
lemda = (3 * 10 ** 8) / (f * 10 ** 6)  # Wavelength
L = lemda / 4  # Antenna Length

print('For (a)')
print('---------')
print(f'Length of the antenna : {L:.3f} m')
print(f'Gain of the antenna : {gain:.1f} = {g:.2f} dB\n')

# Question (b)
print('For (b)')
print('---------')
d = 5000  # T-R separation distance in meters
E0 = 10 ** -3  # Electric-field in V/m
d0 = 1000  # Transmitter distance in meters
ht = 50  # Transmitting antenna height in meters
hr = 1.5  # Receiving antenna height in meters

# Calculating Electric Field
Er_d = (2 * E0 * d0 * 2 * math.pi * ht * hr) / (lemda * d ** 2)  # Electric Field
Ae = (gain * lemda ** 2) / (4 * math.pi)  # Effective Aperture
Pr_d = (Er_d ** 2 / (120 * math.pi)) * Ae  # Received power at distance d
Pr_dB = 10 * math.log10(Pr_d)  # Received power in dBW

print(f'Electric Field, Er(d) : {Er_d:.9f} V/m')
print(f'Effective Aperture, Ae : {Ae:.3f} m^2')
print(f'Received power at 5km distance Er(5 km) : {Pr_dB:.3f} dBW')
```

Output:

```
(For (a)

---------

Length of the antenna : 0.083 m

Gain of the antenna : 1.8 = 2.55 dB


For (b)

---------

Electric Field, Er(d) : 0.000113097 V/m

Effective Aperture, Ae : 0.016 m^2

Received power at 5km distance Er(5 km) : -122.679 dBW
```

Problem No: 10

Name of the Problem: A hexagonal cell within a 4-cell system has a radius of 1.387 km. A total of 60 channels are used within the entire system. If the load per user is 0.029 Erlangs, and λ= call/hour, compute the following for an Erlang C system that has a 5% probability of a delayed call: (a) How many users per square kilometer will this system support? (b) What is the probability that a delayed call will have to wait for more than 10s? (c) What is the probability that a call will be delayed for more than 10 seconds?

Solution:

Given,

Cell radius, R = 1.387 km

Area covered per cell is $2.598 \times (1.387)^2$= 5 sq. km

Number of cells per cluster = 4

Total number of channels = 60

Therefore, number of channels per cell = 60 / 4 = 15 channels.

(a) From Erlang C chart, for 5% probability of delay with C = 15, traffic intensity = 9.0 Erlangs. Therefore, number of users = total traffic intensity / traffic per user

= 9.0/0.029 = 310 users

= 310 users/5 sq. km = 62 users/sq. km

(b) Given $\lambda = 1$, holding time

H = A_u / λ = 0.029 hour = 104.4 seconds.

The probability that a delayed call will have to wait for more than 10 s is

Pr[delay > t|delay] = exp(-(C-A)t/H)

= exp(-(15-9.0)10/104.4) = 56.29 %

(c) Given Pr[delay >0] = 5% = 0.05

Probability that a call is delayed more than 10 seconds,

Pr[delay> 10] = Pr[delay>0]Pr[delay>t|delay]

= 0.05×0.5629 = 2.81 %

Source Code (Python Code):

```
import numpy as np

# Define constants
R = 1.387  # Cell Radius
n = 4  # Number of cells
N = 60  # Total number of channels
area = round(2.5981 * R**2)  # Area covered per cell
C = N / 4  # Number of channels per cell
A = 9  # Traffic intensity at c=15, GOS=0.05, Au=0.029 from Erlang C chart

# Question (a)
Au = 0.029  # Traffic per user
U = int(A / Au)  # Number of users
U_per = round(U / area)  # Number of users per square km
print(f'(a) Number of users per square km: {U_per} users/sq km\n')

# Question (b)
lambda_ = 1  # lambda = 1 hour
H = (Au / lambda_) * 3600  # Holding Time (hour to second)
Prb = np.exp((-(C - A) * 10) / H)  # t=10s, C=15, A=9, H=104.4
print(f'(b) The probability that a delayed call will have to wait: {Prb * 100:.2f}%\n')

# Question (c)
Prc = 0.05 * Prb * 100  # 5% probability of delayed call
print(f'(c) The probability that a call will be delayed: {Prc:.2f}%\n')
```

Output:

```
(a) Number of users per square km: 62 users/sq km

(b) The probability that a delayed call will have to wait:
56.29%

(c) The probability that a call will be delayed: 2.81%
```