

Федеральное агентство связи

СибГУТИ

**Кафедра телекоммуникационных сетей и
вычислительных средств (ТС и ВС)**

Дисциплина

Сети ЭВМ и телекоммуникации 2.0

Лабораторная работа №6

Планирование беспроводных сетей Wi-fi.

Выполнил: студент группы ИА-832

Тиванов.Д.Е

Проверил: преподаватель

Ахпашев Р.В

Новосибирск 2021

Задание

Лабораторная работа состоит из двух этапов:

- ∞ Расчет максимального радиуса покрытия для точки доступа Wi-fi (для двух несущих частот: 2.4 [GHz], 5 [GHz]).
- ∞ Визуализация в виде "тепловой" карты по мощности сигнала.
- ∞ Оптимизация модели распространения радиосигнала на базе ваших

измерений и теоретического расчета.

Ход работы:

Теоретические допущения:

<u>Аа</u> Параметр	<u>≡</u> Точка доступа	<u>≡</u> Пользовательское устройство
<u>TxPower</u> [mWatt].	100, 200, 300, 600	100, 200
<u>Antenna Gain</u> [dBi].	[10 - 22]	[0]
<u>Noise Figure</u> [dB].	[3.5; 5]	[6.5; 8]
<u>Bandwidth</u> [MHz].	[5; 10; 15; 20]	[5; 10; 15; 20]
<u>Req. SINR</u> [dB].	[(-5) - 30]	[(-5) - 30]
<u>Общие параметры</u>	_____	_____
<u>Carriere Frequency</u> [GHz].	[2.4; 5]	

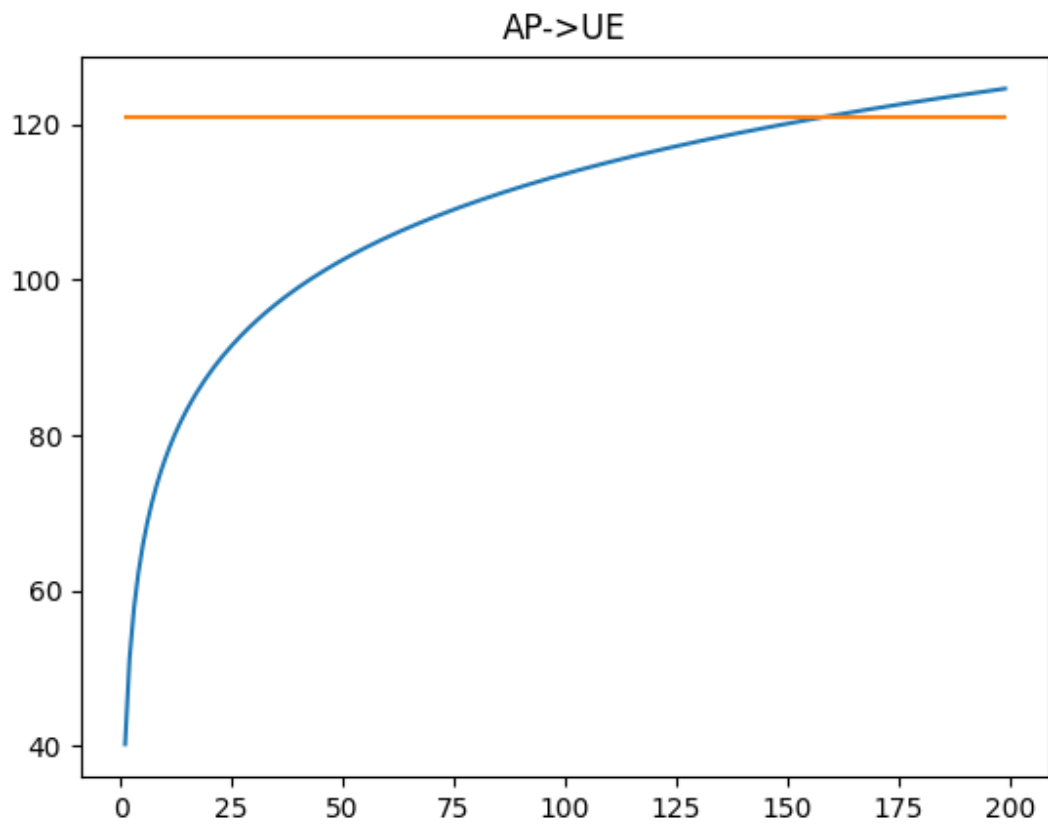
6. Планирование беспроводных сетей Wi-fi.

<u>Аа</u> Параметр	<u>≡</u> Точка доступа	<u>≡</u> Пользовательское устройство
<u>Building.penetration</u> [dB].	[8 - 26]	
<u>Interference Margin</u> [dB].	[3 - 6]	

Максимальные потери от точки доступа до пользовательского устройства :

Figure 1

— □ ×

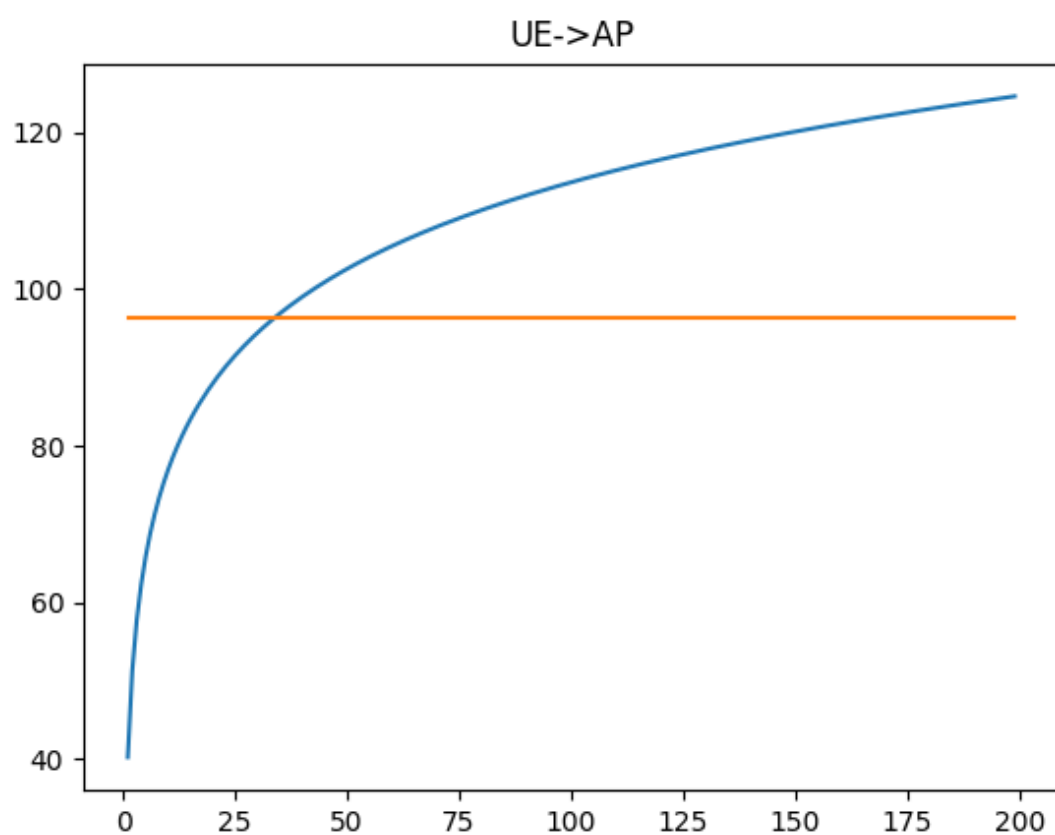
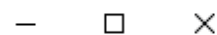


x=-0.1 y=87.6

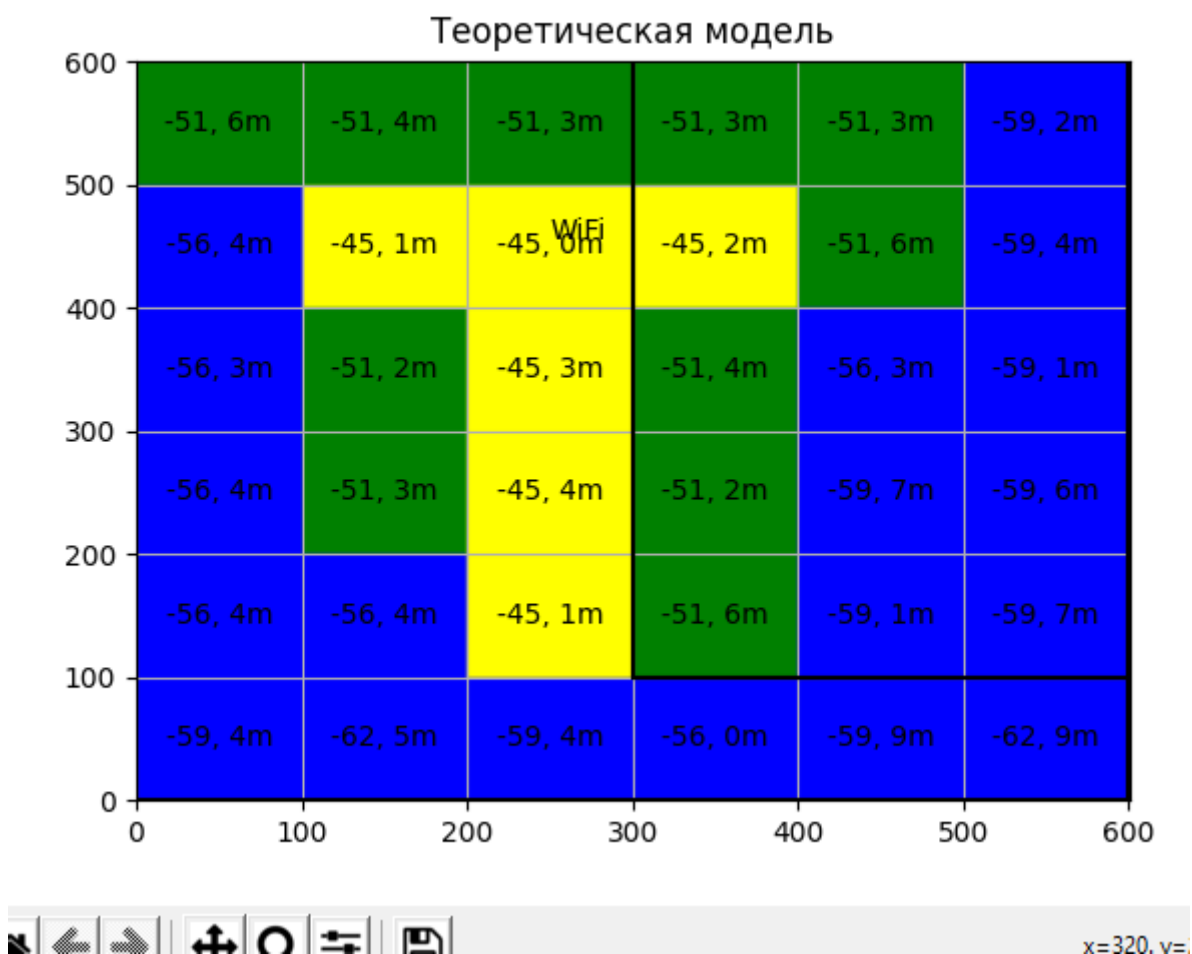
Максимальные потери от пользовательского устройства до точки доступа



Figure 1



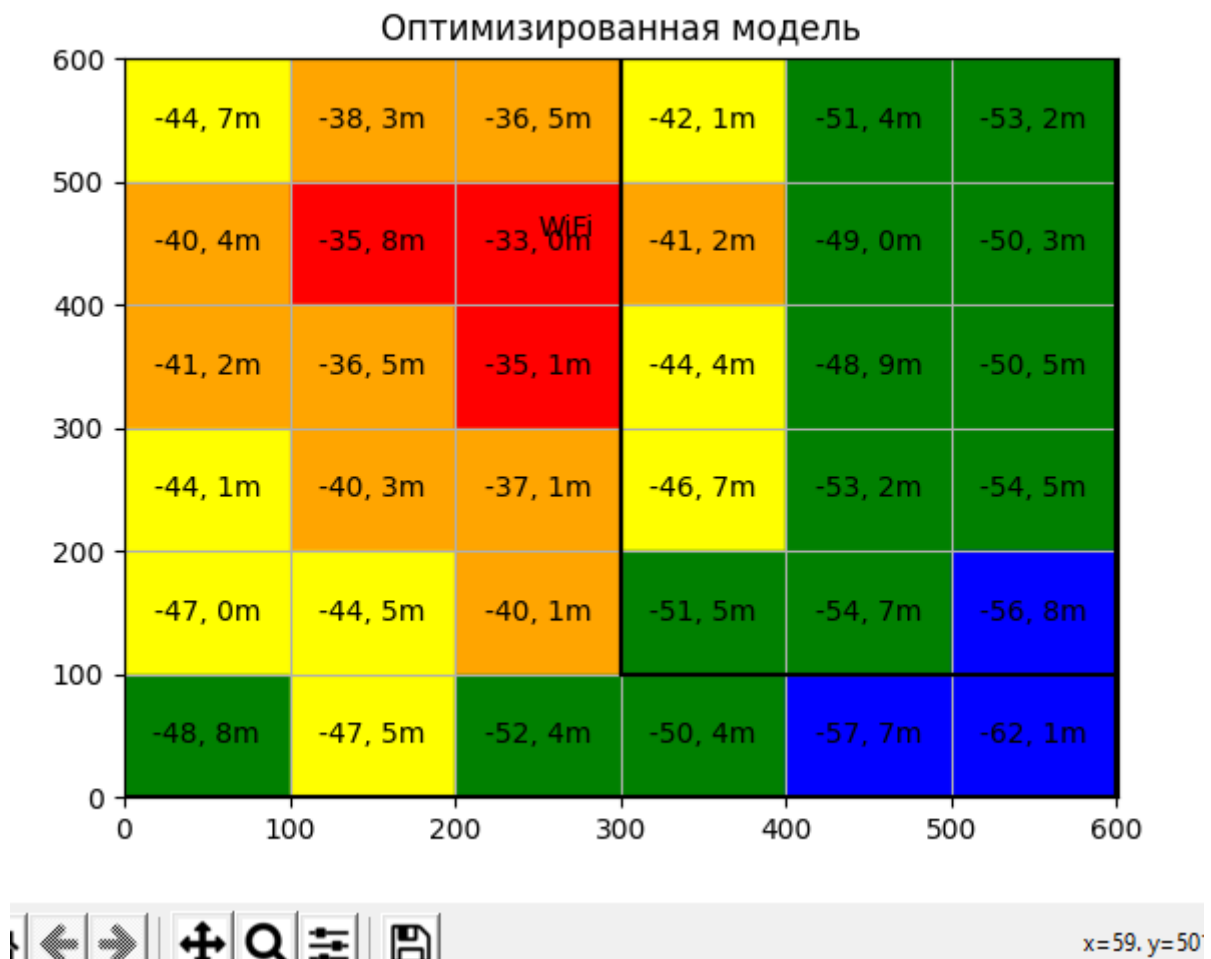
Часть 2. Визуализация "тепловой" карты:



Значение мощности - цвет

<u>Аа</u> Значение мощности [dBm]	<u>≡</u> Цвет
[> -55.]	Красный
[-56; -75].	Оранжевый
[-76; -98].	Желтый
[-99; -115].	Зеленый
[< -116.]	Синий

Часть 3. Оптимизация модели распространения радиосигнала



Код программы:

```
import matplotlib.pyplot as plt
import random
import math
import matplotlib.pyplot as plt

TxPower = [[20, 23.01, 24.77, 27.78], [20, 23.01]]
AP_TxPower = TxPower[0][random.randint(0, 3)]
UE_TxPower = TxPower[1][random.randint(0, 1)]

AP_AntennaGain = random.randint(10, 22)
UE_AntennaGain = 0

AP_NoiseFigure = random.uniform(3.5, 5)
UE_NoiseFigure = random.uniform(6.5, 8)

AP_Bandwidth = random.randrange(5, 25, 5)
```

```

UE_Bandwidth = random.randrange(5, 25, 5)

AP_SINR = random.randint(-2, 12)
UE_SINR = random.randint(-2, 12)

CarriereFrequency = random.uniform(2.4, 5)
BuildingPenetration = random.randint(8, 26)
InterferenceMargin = random.randint(3, 6)

def Pl(distance):
    return 26 * math.log10(CarriereFrequency) + 22.7 + 36.7 *
math.log10(distance)

AP_ThermNoise = -174 + 10 * math.log10(AP_Bandwidth * 10 ** 6)
AP_RxSens = AP_ThermNoise + AP_SINR + AP_NoiseFigure
AP_MAPL = AP_TxPower + AP_AntennaGain - BuildingPenetration -
InterferenceMargin - AP_RxSens
pl = []
x = range(1, 200)
y = []
for d in x:
    pl.append(Pl(d))
    y.append(AP_MAPL)
for d in x:
    if pl[d] >= AP_MAPL:
        AP_dm = d
        break
print("PL(d) = " + str(pl))
print("AP_MAPL = "+str(AP_MAPL))
AP_radius = (1 + AP_dm ** 2) ** (1 / 2)
print("AP_radius = " + str(AP_radius))
plt.plot(x, pl, x, y)
plt.title("AP->UE")
#plt.show()

UE_ThermNoise = -174 + 10 * math.log10(UE_Bandwidth * 10 ** 6)
UE_RxSens = UE_ThermNoise + UE_SINR + UE_NoiseFigure
UE_MAPL = UE_TxPower + UE_AntennaGain - BuildingPenetration -
InterferenceMargin - UE_RxSens
y = []
for d in x:
    y.append(UE_MAPL)
for d in x:
    if pl[d] >= UE_MAPL:
        UE_dm = d
        break
print("UE_MAPL = " + str(UE_MAPL))
UE_radius = (1 + UE_dm ** 2) ** (1 / 2)
print("UE_radius = " + str(UE_radius))
plt.plot(x, pl, x, y)
plt.title("UE->AP")
#plt.show()
plt.close()

def text_on_plot(x, y, value, dist):
    if value >= -35:
        color = 'red'
    elif -41 <= value <= -36:
        color = 'orange'
    elif -47 <= value <= -42:
        color = 'yellow'

```

```

elif -54 <= value <= -48:
    color = 'green'
elif -55 >= value:
    color = 'blue'
ax.fill_between(range(x // 100 * 100, x // 100 * 100 + 100), y // 100 *
100, y // 100 * 100 + 100, color=color)
plt.text(
    x, y, '{0}, {1}m'.format(int(value), dist),
    ha='center', va='center',
    transform=ax.transData
)

fig = plt.figure()
ax = fig.add_subplot(111)
ax.set(xLim=[0, 601],
        yLim=[0, 601])
ax.plot([0, 600, 600, 300, 300, 600, 300, 300, 0, 0], [0, 0, 600, 600, 100,
100, 100, 600, 600, 0], color='black')

text_on_plot(50, 550, -pl[2], 6)
text_on_plot(50, 450, -pl[3], 4)
text_on_plot(50, 350, -pl[3], 3)
text_on_plot(50, 250, -pl[3], 4)
text_on_plot(50, 150, -pl[3], 4)
text_on_plot(50, 50, -pl[4], 4)

text_on_plot(150, 550, -pl[2], 4)
text_on_plot(150, 450, -pl[1], 1)
text_on_plot(150, 350, -pl[2], 2)
text_on_plot(150, 250, -pl[2], 3)
text_on_plot(150, 150, -pl[3], 4)
text_on_plot(150, 50, -pl[5], 5)

plt.text( # position text relative to data
    250, 450, 'WiFi', # x, y, text,
    ha='left', va='bottom', # text alignment,
    transform=ax.transData # coordinate system transformation
)

text_on_plot(250, 550, -pl[2], 3)
text_on_plot(250, 450, -pl[1], 0)
text_on_plot(250, 350, -pl[1], 3)
text_on_plot(250, 250, -pl[1], 4)
text_on_plot(250, 150, -pl[1], 1)
text_on_plot(250, 50, -pl[4], 4)

text_on_plot(350, 550, -pl[2], 3)
text_on_plot(350, 450, -pl[1], 2)
text_on_plot(350, 350, -pl[2], 4)
text_on_plot(350, 250, -pl[2], 2)
text_on_plot(350, 150, -pl[2], 6)
text_on_plot(350, 50, -pl[3], 0)

text_on_plot(450, 550, -pl[2], 3)
text_on_plot(450, 450, -pl[2], 6)
text_on_plot(450, 350, -pl[3], 3)
text_on_plot(450, 250, -pl[4], 7)
text_on_plot(450, 150, -pl[4], 1)
text_on_plot(450, 50, -pl[4], 9)

text_on_plot(550, 550, -pl[4], 2)
text_on_plot(550, 450, -pl[4], 4)

```



```

text_on_plot(550, 350, -pl[4], 1)
text_on_plot(550, 250, -pl[4], 6)
text_on_plot(550, 150, -pl[4], 7)
text_on_plot(550, 50, -pl[5], 9)

plt.grid(True)
plt.title("Теоретическая модель")
plt.show()

fig = plt.figure()
ax = fig.add_subplot(111)
ax.set(xLim=[0, 601],
       yLim=[0, 601])
ax.plot([0, 600, 600, 300, 300, 600, 300, 300, 0, 0], [0, 0, 600, 600, 100,
100, 100, 600, 600, 0], color='black')

text_on_plot(50, 550, -44, 7)
text_on_plot(50, 450, -40, 4)
text_on_plot(50, 350, -41, 2)
text_on_plot(50, 250, -44, 1)
text_on_plot(50, 150, -47, 0)
text_on_plot(50, 50, -48, 8)

text_on_plot(150, 550, -38, 3)
text_on_plot(150, 450, -35, 8)
text_on_plot(150, 350, -36, 5)
text_on_plot(150, 250, -40, 3)
text_on_plot(150, 150, -44, 5)
text_on_plot(150, 50, -47, 5)

plt.text( # position text relative to data
        250, 450, 'WiFi', # x, y, text,
        ha='left', va='bottom', # text alignment,
        transform=ax.transData # coordinate system transformation
)
text_on_plot(250, 550, -36, 5)
text_on_plot(250, 450, -33, 0)
text_on_plot(250, 350, -35, 1)
text_on_plot(250, 250, -37, 1)
text_on_plot(250, 150, -40, 1)
text_on_plot(250, 50, -52, 4)

text_on_plot(350, 550, -42, 1)
text_on_plot(350, 450, -41, 2)
text_on_plot(350, 350, -44, 4)
text_on_plot(350, 250, -46, 7)
text_on_plot(350, 150, -51, 5)
text_on_plot(350, 50, -50, 4)

text_on_plot(450, 550, -51, 4)
text_on_plot(450, 450, -49, 0)
text_on_plot(450, 350, -48, 9)
text_on_plot(450, 250, -53, 2)
text_on_plot(450, 150, -54, 7)
text_on_plot(450, 50, -57, 7)

text_on_plot(550, 550, -53, 2)
text_on_plot(550, 450, -50, 3)
text_on_plot(550, 350, -50, 5)
text_on_plot(550, 250, -54, 5)
text_on_plot(550, 150, -56, 8)

```

```
text_on_plot(550, 50, -62, 1)

plt.grid(True)
plt.title("Оптимизированная модель")
plt.show()
```