

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра теории вероятностей и кибербезопасности

ОТЧЕТ

ПО ПРАКТИЧЕСКОЙ РАБОТЕ №2

дисциплина: Построение и анализ моделей беспроводных сетей 5G/6G

Студент: Быстров Глеб Андреевич

Группа: НФИмд-01-24

Преподаватель: Бегишев Вячеслав Олегович

МОСКВА

2024 г.

Цель работы:

Изучение моделей потерь сигнала в различных условиях распространения радиоволн, таких как городская среда (Urban Macro, LOS/NLOS) и внутри зданий (Indoor, LOS/NLOS). Проведение анализа влияния различных параметров (мощности передатчика, усиления антенн, уровня шума, расстояния и частоты) на значения отношения сигнал/шум (SNR) и пропускной способности канала связи, рассчитанных по формуле Шеннона.

Список сокращений:

- FSPL — Free Space Path Loss (Потери на свободном пространстве)
- LoS — Line of Sight (Прямая видимость)
- nLoS — Non-Line of Sight (Непрямая видимость)
- UMa — Urban Macro (Городская макросреда)
- InH — Indoor (Внутренние помещения, офисная среда)

Выполнение работы

Задание.

Отношение SNR

1. Приведите формулу зависимости отношения уровня полезного сигнала к шуму (Signal-to-Noise ratio, SNR) для приемника, находящегося на расстоянии от передатчика с применением модели распространения FSPL.
2. Постройте графики значений отношения SNR в зависимости от расстояния между передатчиком и приемником для частот $f_1 = 900$ МГц, $f_2 = 1.9$ ГГц и $f_3 = 28$ ГГц. Излучаемая мощность антенны передатчика – 23 дБм, усиления антенн на передаче и приеме равны 10 дБ. Ширина полосы пропускания радиоканала – 20 МГц, тепловой шум – 174 дБ/Гц.
3. Что будет являться уровнем уверенного приема в данном случае? Определите максимально возможную дистанцию уверенного приема.
4. Что произойдет с максимальной дистанцией связи если изменять чувствительность приемника, усиления антенн, излучаемую мощность?
5. Как изменится формула если добавить интерференцию от других передатчиков на той же частоте?

```
import numpy as np
import math
from scipy.constants import speed_of_light as c
import matplotlib.pyplot as plt

def log(value):
    return 10 * np.log10(value)

def lin(value):
    return np.power(10, value / 10)

def fspl(d, f, h_bs, h_ut):
    loss = np.power((4*np.pi*d*f)/c, 2)
    loss = log(loss)
    return loss

def prd(G_r, G_t, P_t, d, f, N_0, B, loss=fspl, h_bs=None, h_ut=None):
    power_on_reciever = log(lin(P_t) * G_t * G_r) - loss(d, f, h_bs, h_ut)
    snr = power_on_reciever - log(B*lin(N_0))
```

```

power_on_reciever = log(lin(P_t) * G_t * G_r) - loss(d, f, h_bs, h_ut)
snr = power_on_reciever - log(B*lin(N_0))
return power_on_reciever, snr

P_t = 23 # мощность передатчика
G_t, G_r = 10, 10 # усиление
N_0 = -174 # шум
B = 20*(10**6) # полоса частот
f = np.array([900*(10**6), 1.9*(10**9), 28*(10**9)]) # частоты
d = np.linspace(0.1, 10000, 100*100) # расстояние

pwr_900, snr_900 = prd(G_r, G_t, P_t, d, f[0], N_0, B)
pwr_1900, snr_1900 = prd(G_r, G_t, P_t, d, f[1], N_0, B)
pwr_28000, snr_28000 = prd(G_r, G_t, P_t, d, f[2], N_0, B)

d_snr_900_filtered = []
snr_900_filtered = []

d_snr_1900_filtered = []
snr_1900_filtered = []

d_snr_28000_filtered = []
snr_28000_filtered = []

filt = -65

for i in range(len(pwr_900)):
    if pwr_900[i] > filt:
        snr_900_filtered.append(snr_900[i])
        d_snr_900_filtered.append(d[i])
    if pwr_1900[i] > filt:
        snr_1900_filtered.append(snr_1900[i])
        d_snr_1900_filtered.append(d[i])
    if pwr_28000[i] > filt:
        snr_28000_filtered.append(snr_28000[i])
        d_snr_28000_filtered.append(d[i])

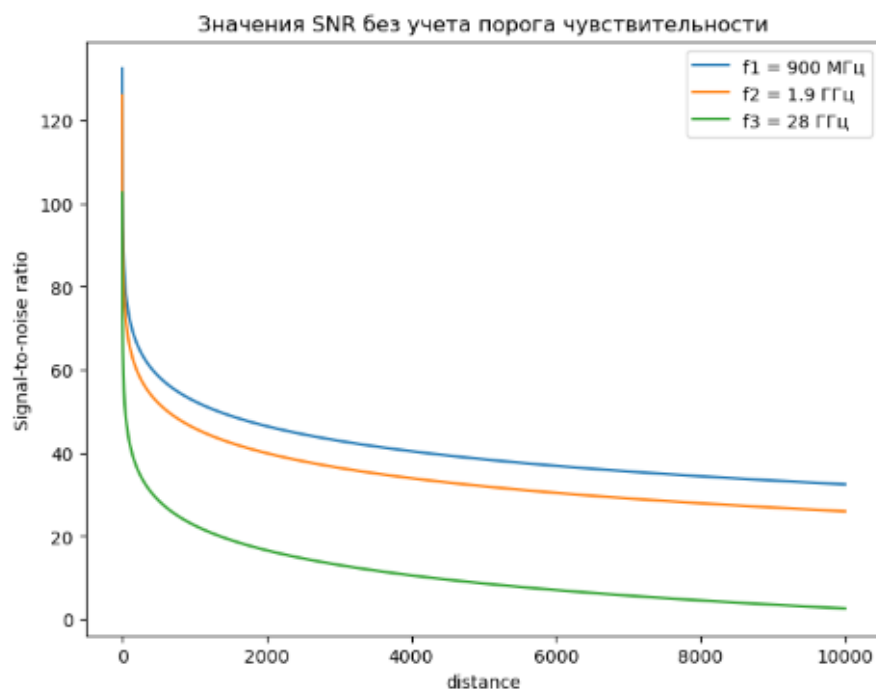
plt.figure(figsize=(8,6))
plt.plot(d, snr_900, label='f1 = 900 МГц')
plt.plot(d, snr_1900, label='f2 = 1.9 ГГц')
plt.plot(d, snr_28000, label='f3 = 28 ГГц')
plt.title("Значения SNR без учета порога чувствительности")
plt.xlabel('distance')
plt.ylabel('Signal-to-noise ratio')
plt.legend()
plt.show()

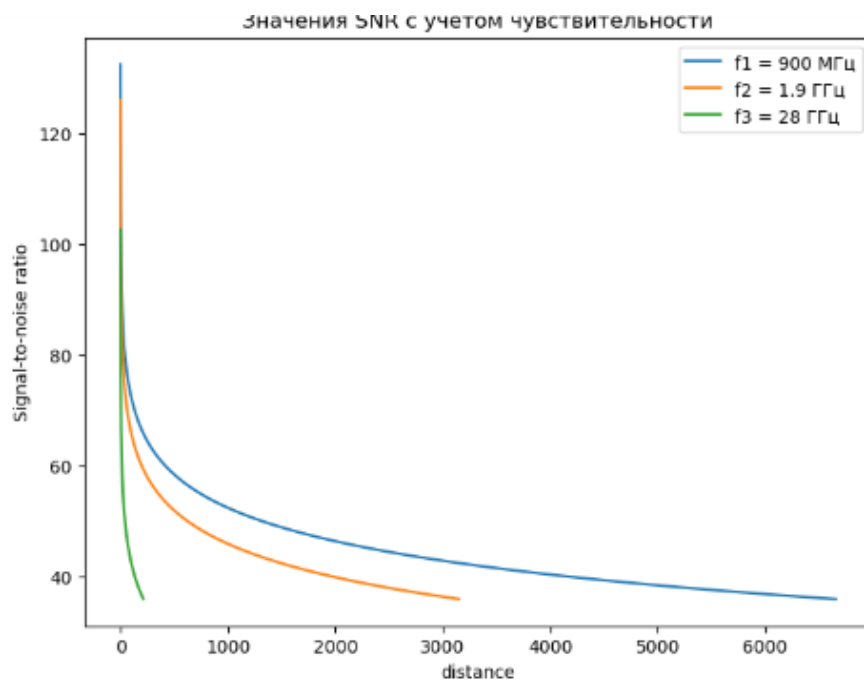
```

```

plt.figure(figsize=(8,6))
plt.plot(d_snr_900_filtered, snr_900_filtered, label='f1 = 900 МГц')
plt.plot(d_snr_1900_filtered, snr_1900_filtered, label='f2 = 1.9 ГГц')
plt.plot(d_snr_28000_filtered, snr_28000_filtered, label='f3 = 28 ГГц')
plt.title("Значения SNR с учетом чувствительности")
plt.xlabel('distance')
plt.ylabel('Signal-to-noise ratio')
plt.legend()
plt.show()

```





```
d_snr_900_filtered = []
snr_900_filtered = []

d_snr_1900_filtered = []
snr_1900_filtered = []

d_snr_28000_filtered = []
snr_28000_filtered = []

filt = -75

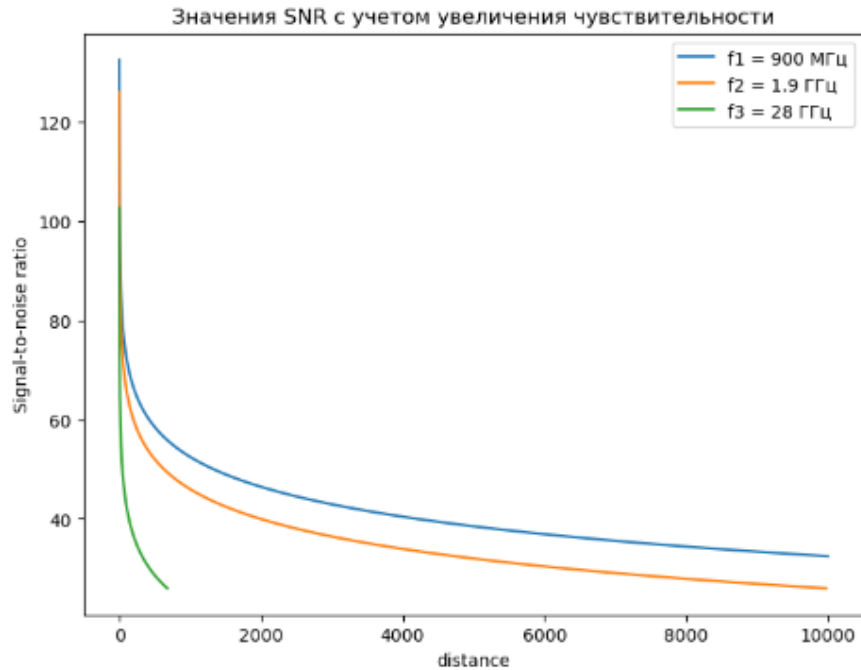
for i in range(len(pwr_900)):
    if pwr_900[i] > filt:
```

```

if pwr_28000[i] > filt:
    snr_28000_filtered.append(snr_28000[i])
    d_snr_28000_filtered.append(d[i])

plt.figure(figsize=(8,6))
plt.plot(d_snr_900_filtered, snr_900_filtered, label='f1 = 900 МГц')
plt.plot(d_snr_1900_filtered, snr_1900_filtered, label='f2 = 1.9 ГГц')
plt.plot(d_snr_28000_filtered, snr_28000_filtered, label='f3 = 28 ГГц')
plt.title("Значения SNR с учетом увеличения чувствительности")
plt.xlabel('distance')
plt.ylabel('Signal-to-noise ratio')
plt.legend()
plt.show()

```



```

P_t = 35 # мощность передатчика
G_t, G_r = 10, 10 # усиление
N_0 = -174 # шум
B = 20*(10**6) # полоса частот
d = np.linspace(0.1, 10000, 100*100) # расстояние

pwr_900, snr_900 = prd(G_r, G_t, P_t, d, f[0], N_0, B)
pwr_1900, snr_1900 = prd(G_r, G_t, P_t, d, f[1], N_0, B)
pwr_28000, snr_28000 = prd(G_r, G_t, P_t, d, f[2], N_0, B)

d_snr_900_filtered = []
snr_900_filtered = []

d_snr_1900_filtered = []
snr_1900_filtered = []

d_snr_28000_filtered = []
snr_28000_filtered = []

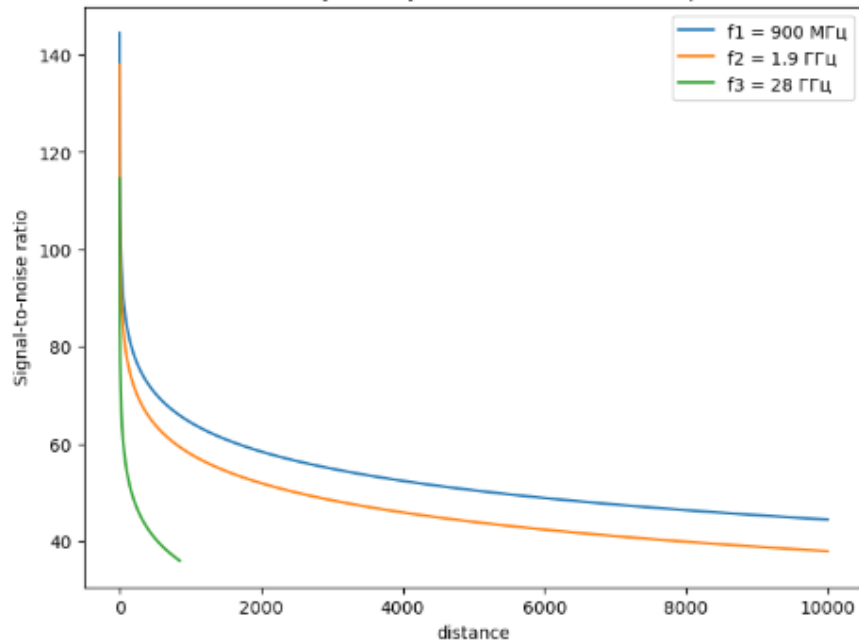
filt = -65

for i in range(len(pwr_900)):
    if pwr_900[i] > filt:
        snr_900_filtered.append(snr_900[i])
        d_snr_900_filtered.append(d[i])
    if pwr_1900[i] > filt:
        snr_1900_filtered.append(snr_1900[i])
        d_snr_1900_filtered.append(d[i])
    if pwr_28000[i] > filt:
        snr_28000_filtered.append(snr_28000[i])
        d_snr_28000_filtered.append(d[i])

plt.figure(figsize=(8,6))
plt.plot(d_snr_900_filtered, snr_900_filtered, label='f1 = 900 МГц')
plt.plot(d_snr_1900_filtered, snr_1900_filtered, label='f2 = 1.9 ГГц')
plt.plot(d_snr_28000_filtered, snr_28000_filtered, label='f3 = 28 ГГц')
plt.title("Значения SNR с учетом увеличения мощности передатчика")
plt.xlabel('distance')
plt.ylabel('Signal-to-noise ratio')
plt.legend()
plt.show()

```

Значения SNR с учетом увеличения мощности передатчика



```

P_t = 23 # мощность передатчика
G_t, G_r = 35, 20 # усиление
N_0 = -174 # шум
B = 20*(10**6) # полоса частот
d = np.linspace(0.1, 10000, 100*100) # расстояние

pwr_900, snr_900 = prd(G_r, G_t, P_t, d, f[0], N_0, B)
pwr_1900, snr_1900 = prd(G_r, G_t, P_t, d, f[1], N_0, B)
pwr_28000, snr_28000 = prd(G_r, G_t, P_t, d, f[2], N_0, B)

d_snr_900_filtered = []
snr_900_filtered = []

```

```

d_snr_900_filtered = []
snr_900_filtered = []

d_snr_1900_filtered = []
snr_1900_filtered = []

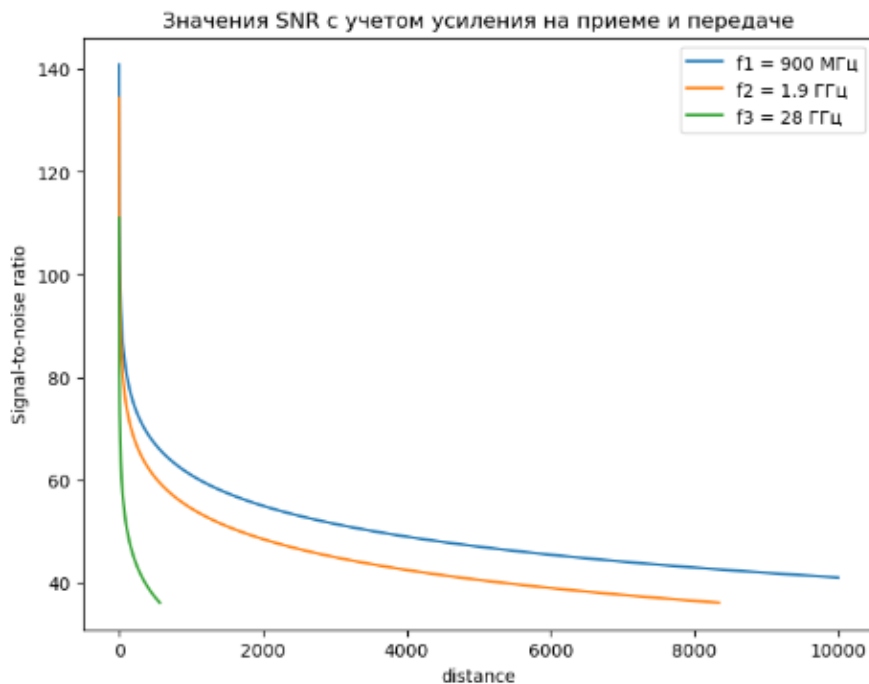
d_snr_28000_filtered = []
snr_28000_filtered = []

filt = -65

for i in range(len(pwr_900)):
    if pwr_900[i] > filt:
        snr_900_filtered.append(snr_900[i])
        d_snr_900_filtered.append(d[i])
    if pwr_1900[i] > filt:
        snr_1900_filtered.append(snr_1900[i])
        d_snr_1900_filtered.append(d[i])
    if pwr_28000[i] > filt:
        snr_28000_filtered.append(snr_28000[i])
        d_snr_28000_filtered.append(d[i])

plt.figure(figsize=(8,6))
plt.plot(d_snr_900_filtered, snr_900_filtered, label='f1 = 900 МГц')
plt.plot(d_snr_1900_filtered, snr_1900_filtered, label='f2 = 1.9 ГГц')
plt.plot(d_snr_28000_filtered, snr_28000_filtered, label='f3 = 28 ГГц')
plt.title("Значения SNR с учетом усиления на приеме и передаче")
plt.xlabel('distance')
plt.ylabel('Signal-to-noise ratio')
plt.legend()
plt.show()

```



Скорость по Шеннону

- В условиях предыдущей задачи, постройте графики теоретической максимальной пропускной способности канала по теореме Шеннона от расстояния между приемником и передатчиком.
- Почему скорость ненулевая даже для дистанций, на которых связь невозможна?
- Зафиксировав дистанцию передачи, постройте графики зависимости пропускной способности от частоты, усиления антенны на передаче, мощности передатчика, и мощности шума.
- За счет чего наиболее эффективно наращивать скорость беспроводных каналов связи?

$$C = B \log_2 \left(1 + \frac{S}{N} \right),$$

где C – пропускная способность канала связи, B – ширина полосы пропускания канала, S – мощность полезного сигнала, N – мощность шума. Обратите внимание, что все величины должны быть в линейной шкале.

```
def shennon(snr, B):
    return B*np.log2(1 + snr)

P_t = 23 # мощность передатчика
G_t, G_r = 10, 10 # усиление
N_0 = -174 # шум
B = 20*(10**6) # полоса частот
d = np.linspace(0.1, 10000, 100*100) # расстояние

pwr_900, snr_900 = prd(G_r, G_t, P_t, d, f[0], N_0, B)
pwr_1900, snr_1900 = prd(G_r, G_t, P_t, d, f[1], N_0, B)
pwr_28000, snr_28000 = prd(G_r, G_t, P_t, d, f[2], N_0, B)

shennon_900 = shennon(ln(snr_900), B)
shennon_1900 = shennon(ln(snr_1900), B)
shennon_28000 = shennon(ln(snr_28000), B)

plt.figure(figsize=(8,6))
plt.plot(d, shennon_900, label='f1 = 900 МГц')
plt.plot(d, shennon_1900, label='f2 = 1.9 ГГц')
plt.plot(d, shennon_28000, label='f3 = 28 ГГц')
plt.xlabel('distance')
plt.ylabel('The Shannon Formula')
```



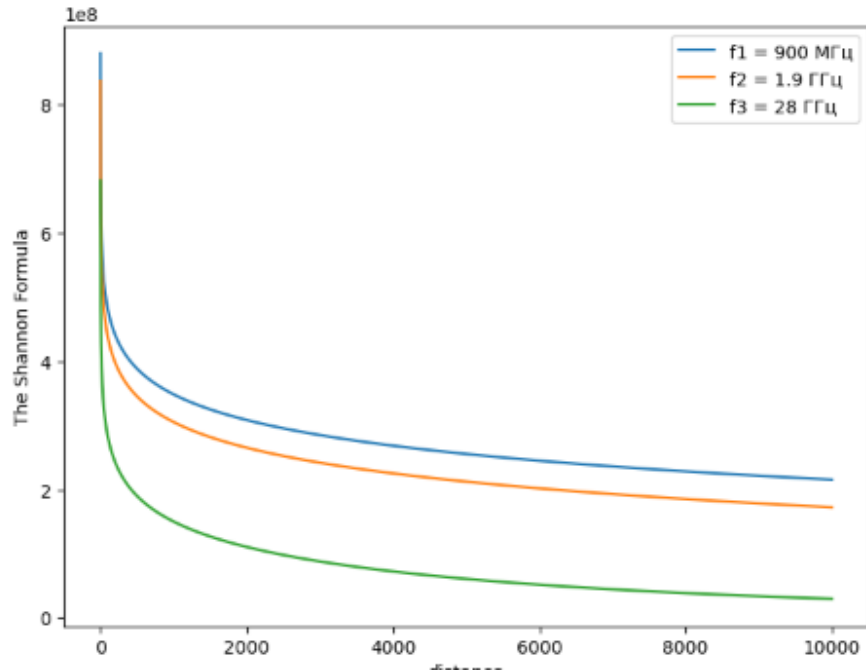
```

pwr_1900, snr_1900 = prd(G_r, G_t, P_t, d, f[1], N_0, B)
pwr_28000, snr_28000 = prd(G_r, G_t, P_t, d, f[2], N_0, B)

shennon_900 = shennon(ln(snr_900), B)
shennon_1900 = shennon(ln(snr_1900), B)
shennon_28000 = shennon(ln(snr_28000), B)

plt.figure(figsize=(8,6))
plt.plot(d, shennon_900, label='f1 = 900 МГц')
plt.plot(d, shennon_1900, label='f2 = 1.9 ГГц')
plt.plot(d, shennon_28000, label='f3 = 28 ГГц')
plt.xlabel('distance')
plt.ylabel('The Shannon Formula')
plt.legend()
plt.show()

```



```

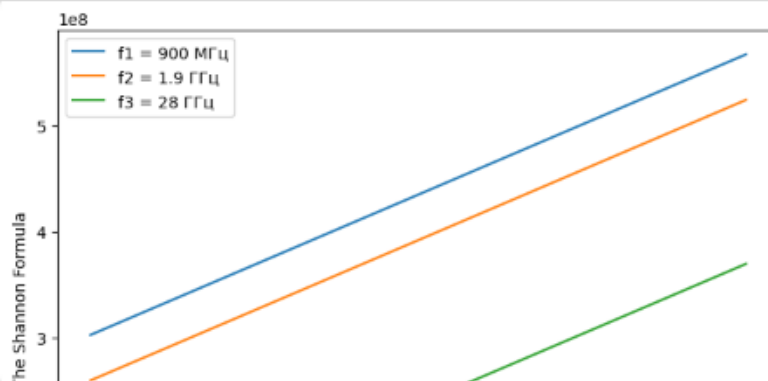
Ввод [156]: P_t = np.linspace(10, 50, 100) # мощность
G_t, G_r = 10, 10 # усиление
N_0 = -174 # шум
B = 20*(10**6) # полоса частот
d = 500

pwr_900, snr_900 = prd(G_r, G_t, P_t, d, f[0], N_0, B)
pwr_1900, snr_1900 = prd(G_r, G_t, P_t, d, f[1], N_0, B)
pwr_28000, snr_28000 = prd(G_r, G_t, P_t, d, f[2], N_0, B)

shennon_900 = shennon(ln(snr_900), B)
shennon_1900 = shennon(ln(snr_1900), B)
shennon_28000 = shennon(ln(snr_28000), B)

plt.figure(figsize=(8,6))
plt.plot(P_t, shennon_900, label='f1 = 900 МГц')
plt.plot(P_t, shennon_1900, label='f2 = 1.9 ГГц')
plt.plot(P_t, shennon_28000, label='f3 = 28 ГГц')
plt.xlabel('power')
plt.ylabel('The Shannon Formula')
plt.legend()
plt.show()

```



```

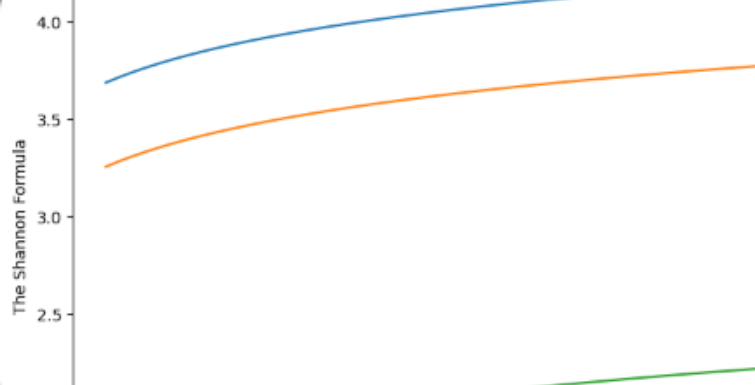
Ввод [157]: P_t = 23 # мощность
G_t, G_r = np.linspace(5, 30, 50), 10 # усиление
N_0 = -174 # шум
B = 20*(10**6) # полоса частот
d = 500

pwr_900, snr_900 = prd(G_r, G_t, P_t, d, f[0], N_0, B)
pwr_1900, snr_1900 = prd(G_r, G_t, P_t, d, f[1], N_0, B)
pwr_28000, snr_28000 = prd(G_r, G_t, P_t, d, f[2], N_0, B)

shannon_900 = shannon(ln(snr_900), B)
shannon_1900 = shannon(ln(snr_1900), B)
shannon_28000 = shannon(ln(snr_28000), B)

plt.figure(figsize=(8,6))
plt.plot(G_t, shannon_900, label='f1 = 900 МГц')
plt.plot(G_t, shannon_1900, label='f2 = 1.9 ГГц')
plt.plot(G_t, shannon_28000, label='f3 = 28 ГГц')
plt.xlabel('gain')
plt.ylabel('The Shannon Formula')
plt.legend()
plt.show()

```



```

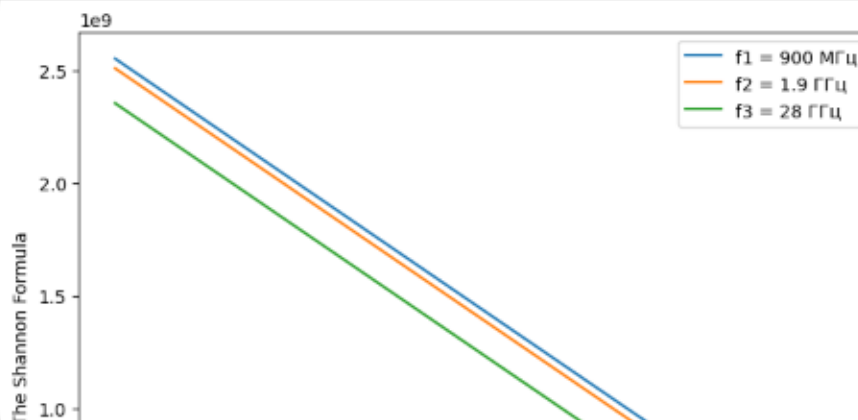
P_t = 23 # мощность
G_t, G_r = 10, 10 # усиление
N_0 = np.linspace(170, 500, 50) # шум
N_0 = -1*N_0
B = 20*(10**6) # полоса частот
d = 500

pwr_900, snr_900 = prd(G_r, G_t, P_t, d, f[0], N_0, B)
pwr_1900, snr_1900 = prd(G_r, G_t, P_t, d, f[1], N_0, B)
pwr_28000, snr_28000 = prd(G_r, G_t, P_t, d, f[2], N_0, B)

shannon_900 = shannon(ln(snr_900), B)
shannon_1900 = shannon(ln(snr_1900), B)
shannon_28000 = shannon(ln(snr_28000), B)

plt.figure(figsize=(8,6))
plt.plot(N_0, shannon_900, label='f1 = 900 МГц')
plt.plot(N_0, shannon_1900, label='f2 = 1.9 ГГц')
plt.plot(N_0, shannon_28000, label='f3 = 28 ГГц')
plt.xlabel('Noise')
plt.ylabel('The Shannon Formula')
plt.legend()
plt.show()

```



Стандартные модели распространения 3GPP

10. В условиях предыдущей задачи, постройте графики пропускной способности по Шеннону от расстояния для различных моделей распространения: FSPL, UMa LoS, UMa nLoS, InH-Office LoS, InH-Office nLoS из стандарта 3GPP TR 38.901.
11. В каких условиях достижимая скорость выше?

UMa LoS

LoS	$PL_{\text{UMa-LoS}} = \begin{cases} PL_1 & 10\text{m} \leq d_{3D} \leq d'_{\text{RP}} \\ PL_2 & d'_{\text{RP}} \leq d_{3D} \leq 5\text{km}, \text{ see note 1} \end{cases}$ $PL_1 = 28.0 + 22 \log_{10}(d_{3D}) + 20 \log_{10}(f_c)$ $PL_2 = 28.0 + 40 \log_{10}(d_{3D}) + 20 \log_{10}(f_c) - 9 \log_{10}((d'_{\text{RP}})^2 + (h_{\text{BS}} - h_{\text{UT}})^2)$	$\sigma_{\text{SF}} = 4$	$1.5\text{m} \leq h_{\text{UT}} \leq 22.5\text{m}$ $h_{\text{BS}} = 25\text{m}$
-----	--	--------------------------	--

UMa nLoS

nLoS	$PL_{\text{UMa-nLoS}} = \max(PL_{\text{UMa-LoS}}, PL'_{\text{UMa-nLoS}})$ <p>for $10\text{m} \leq d_{3D} \leq 5\text{km}$</p> $PL'_{\text{UMa-nLoS}} = 13.54 + 39.08 \log_{10}(d_{3D}) + 20 \log_{10}(f_c) - 0.6(h_{\text{UT}} - 1.5)$	$\sigma_{\text{SF}} = 6$	$1.5\text{m} \leq h_{\text{UT}} \leq 22.5\text{m}$ $h_{\text{BS}} = 25\text{m}$ Explanations: see note 3
	Optional $PL = 32.4 + 20 \log_{10}(f_c) + 30 \log_{10}(d_{3D})$	$\sigma_{\text{SF}} = 7.8$	

InH-Office LoS

LoS	$PL_{\text{InH-LoS}} = 32.4 + 17.3 \log_{10}(d_{3D}) + 20 \log_{10}(f_c)$	$\sigma_{\text{SF}} = 3$	$1\text{m} \leq d_{3D} \leq 150\text{m}$
-----	---	--------------------------	--

InH-Office nLoS

NLOS	$PL_{\text{InH-NLOS}} = \max(PL_{\text{InH-NLOS}}, PL'_{\text{InH-NLOS}})$	$\sigma_{\text{SF}} = 8.03$	$1\text{m} \leq d_{3D} \leq 150\text{m}$
	$PL'_{\text{InH-NLOS}} = 38.3 \log_{10}(d_{3D}) + 17.30 + 24.9 \log_{10}(f_c)$		
	Optional $PL'_{\text{InH-NLOS}} = 32.4 + 20 \log_{10}(f_c) + 31.9 \log_{10}(d_{3D})$	$\sigma_{\text{SF}} = 8.29$	$1\text{m} \leq d_{3D} \leq 150\text{m}$

```
def d_2D_3D(d_2D, h_bs, h_ut):
    return np.sqrt(np.power(d_2D, 2) + np.power(h_bs - h_ut, 2))

def d_3D_2D(d_3D, h_bs, h_ut):
    return np.sqrt(np.power(d_3D, 2) - np.power(h_bs - h_ut, 2))

def uma_los(d_2D, f, h_bs, h_ut):
    def pl_choose(d_2, d_BP, d_3, f):
        if d_2 <= d_BP:
            return 28.0 + 22*np.log10(d_3) + 20*np.log10(f)
        else:
            return 28.0 + 40*np.log10(d_3) + 20*np.log10(f) - 9*np.log10(np.power(d_BP, 2) + np.power(h_bs - h_ut, 2))

    d_3D = d_2D_3D(d_2D, h_bs, h_ut)
    d_BP = 4 * (h_bs - 1) * (h_ut - 1) * f / c
    path_loss = np.array([pl_choose(d_[0], d_BP, d_[1], f) for d_ in zip(d_2D, d_3D)])

    return path_loss

def uma_nlos(d_2D, f, h_bs, h_ut):
    d_3D = d_2D_3D(d_2D, h_bs, h_ut)
    uma_nlos_ = 13.54 + 39.08*np.log10(d_3D) + 20*np.log10(f) - 0.6*np.log10(h_ut - 1.5)
    uma_los_ = uma_los(d_2D, f, h_bs, h_ut)
    path_loss = np.array([max(u[0], u[1]) for u in zip(uma_los_, uma_nlos_)])

    return path_loss

def inh_los(d_2D, f, h_bs, h_ut):
    d_3D = d_2D_3D(d_2D, h_bs, h_ut)
    path_loss = 32.4 + 17.3*np.log10(d_3D) + 20*np.log10(f)

    return path_loss

def inh_nlos(d_2D, f, h_bs, h_ut):
    d_3D = d_2D_3D(d_2D, h_bs, h_ut)
    inh_nlos = 38.3*np.log10(d_3D) + 17.30 + 20*np.log10(f)
```

```

    return path_loss

def inh_los(d_2D, f, h_bs, h_ut):
    d_3D = d_2D_3D(d_2D, h_bs, h_ut)
    path_loss = 32.4 + 17.3*np.log10(d_3D) + 20*np.log10(f)

    return path_loss

def inh_nlos(d_2D, f, h_bs, h_ut):
    d_3D = d_2D_3D(d_2D, h_bs, h_ut)
    inh_nlos_ = 38.3*np.log10(d_3D) + 17.30 + 20*np.log10(f)
    inh_los_ = inh_los(d_2D, f, h_bs, h_ut)
    path_loss = np.array([max(i[0], i[1]) for i in zip(inh_los_, inh_nlos_)])

    return path_loss

d_uma = np.linspace(10, 5000, 10000)
h_ut_uma = 12
h_bs_uma = 25

h_ut_inh = 1
h_bs_inh = 3
d_2D_max = d_2D_3D(150, h_bs_inh, h_ut_inh)
d_inh = np.linspace(0, d_2D_max, 10000)

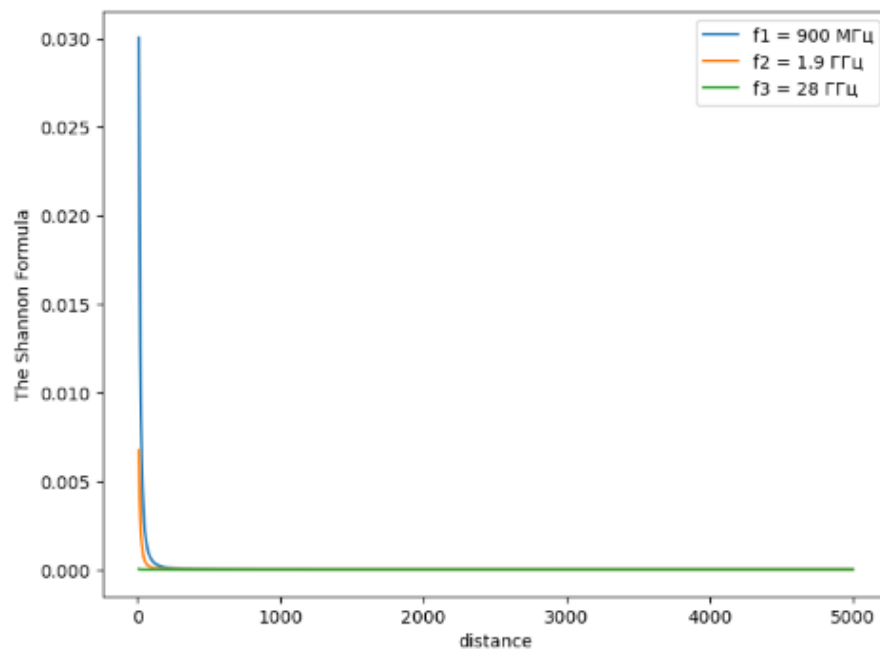
P_t = 23 # мощность передатчика
G_t, G_r = 10, 10 # усиление
N_0 = -174 # шум
B = 20*(10**6) # полоса частот

pwr_900, snr_900 = prd(G_r, G_t, P_t, d_uma, f[0], N_0, B, loss=uma_los, h_bs=h_bs_uma, h_ut=h_ut_uma)
pwr_1900, snr_1900 = prd(G_r, G_t, P_t, d_uma, f[1], N_0, B, loss=uma_los, h_bs=h_bs_uma, h_ut=h_ut_uma)
pwr_28000, snr_28000 = prd(G_r, G_t, P_t, d_uma, f[2], N_0, B, loss=uma_los, h_bs=h_bs_uma, h_ut=h_ut_uma)

shannon_900 = shannon(ln(snr_900), B)
shannon_1900 = shannon(ln(snr_1900), B)
shannon_28000 = shannon(ln(snr_28000), B)

plt.figure(figsize=(8,6))
plt.plot(d_uma, shannon_900, label='f1 = 900 МГц')
plt.plot(d_uma, shannon_1900, label='f2 = 1.9 ГГц')
plt.plot(d_uma, shannon_28000, label='f3 = 28 ГГц')
plt.xlabel('distance')
plt.ylabel('The Shannon Formula')
plt.legend()
plt.show()

```



```

: pwr_900, snr_900 = prd(G_r, G_t, P_t, d_uma, f[0], N_0, B, loss=uma_nlos, h_bs=h_bs_uma, h_ut=h_ut_uma)
  pwr_1900, snr_1900 = prd(G_r, G_t, P_t, d_uma, f[1], N_0, B, loss=uma_nlos, h_bs=h_bs_uma, h_ut=h_ut_uma)
  pwr_28000, snr_28000 = prd(G_r, G_t, P_t, d_uma, f[2], N_0, B, loss=uma_nlos, h_bs=h_bs_uma, h_ut=h_ut_uma)

  shennond_900 = shennon(ln(snr_900), B)
  shennon_1900 = shennon(ln(snr_1900), B)
  shennon_28000 = shennon(ln(snr_28000), B)

  plt.figure(figsize=(8,6))
  plt.plot(d_uma, shennon_900, label='f1 = 900 МГц')
  plt.plot(d_uma, shennon_1900, label='f2 = 1.9 ГГц')
  plt.plot(d_uma, shennon_28000, label='f3 = 28 ГГц')

```

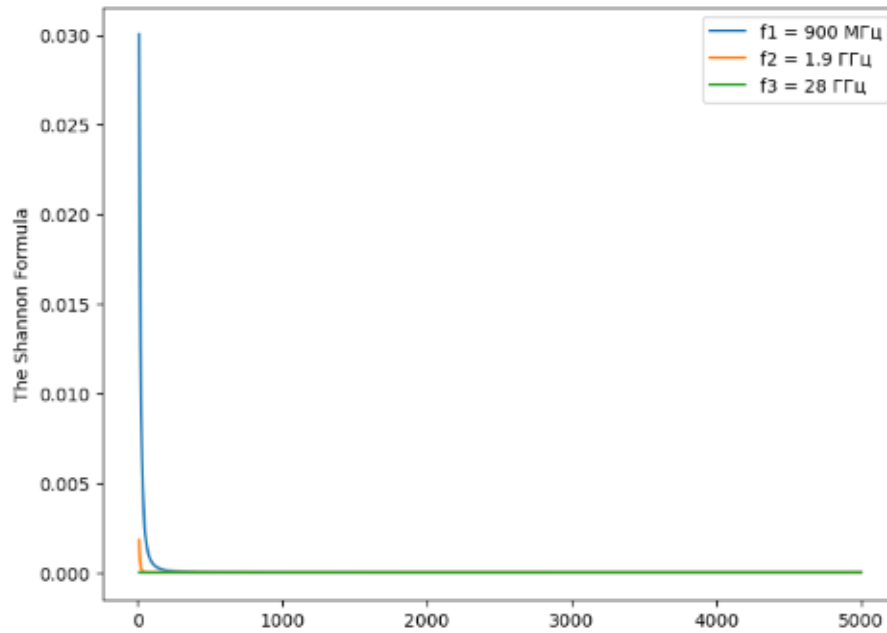
```

pwr_900, snr_900 = prd(G_r, G_t, P_t, d_uma, f[0], N_0, B, loss=uma_nlos, h_bs=h_bs_uma, h_ut=h_ut_uma)
pwr_1900, snr_1900 = prd(G_r, G_t, P_t, d_uma, f[1], N_0, B, loss=uma_nlos, h_bs=h_bs_uma, h_ut=h_ut_uma)
pwr_28000, snr_28000 = prd(G_r, G_t, P_t, d_uma, f[2], N_0, B, loss=uma_nlos, h_bs=h_bs_uma, h_ut=h_ut_uma)

shennond_900 = shennon(ln(snr_900), B)
shennond_1900 = shennon(ln(snr_1900), B)
shennond_28000 = shennon(ln(snr_28000), B)

plt.figure(figsize=(8,6))
plt.plot(d_uma, shennond_900, label='f1 = 900 МГц')
plt.plot(d_uma, shennond_1900, label='f2 = 1.9 ГГц')
plt.plot(d_uma, shennond_28000, label='f3 = 28 ГГц')
plt.xlabel('distance')
plt.ylabel('The Shannon Formula')
plt.legend()
plt.show()

```

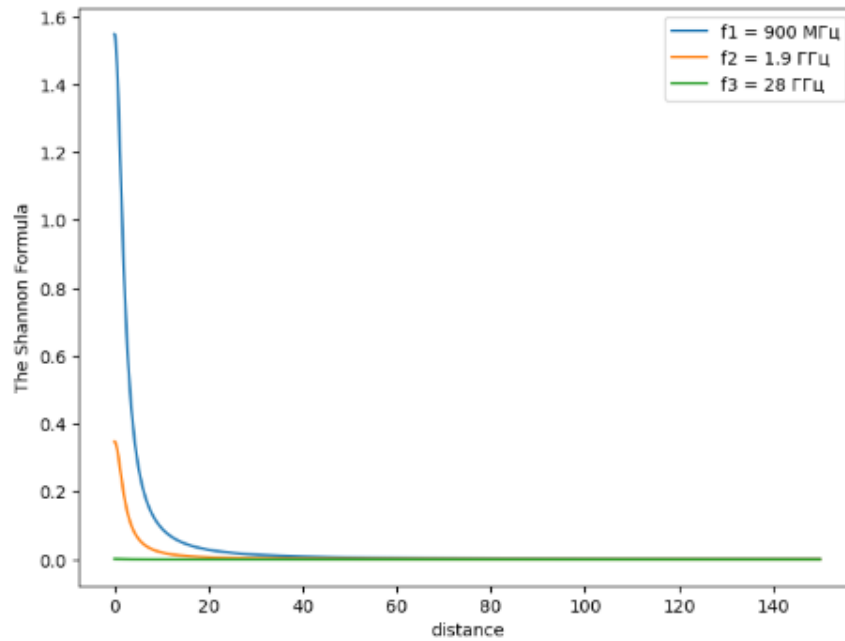


```

shennon_900 = shennon(ln(snr_900), B)
shennon_1900 = shennon(ln(snr_1900), B)
shennon_28000 = shennon(ln(snr_28000), B)

plt.figure(figsize=(8,6))
plt.plot(d_inh, shennon_900, label='f1 = 900 МГц')
plt.plot(d_inh, shennon_1900, label='f2 = 1.9 ГГц')
plt.plot(d_inh, shennon_28000, label='f3 = 28 ГГц')
plt.xlabel('distance')
plt.ylabel('The Shannon Formula')
plt.legend()
plt.show()

```



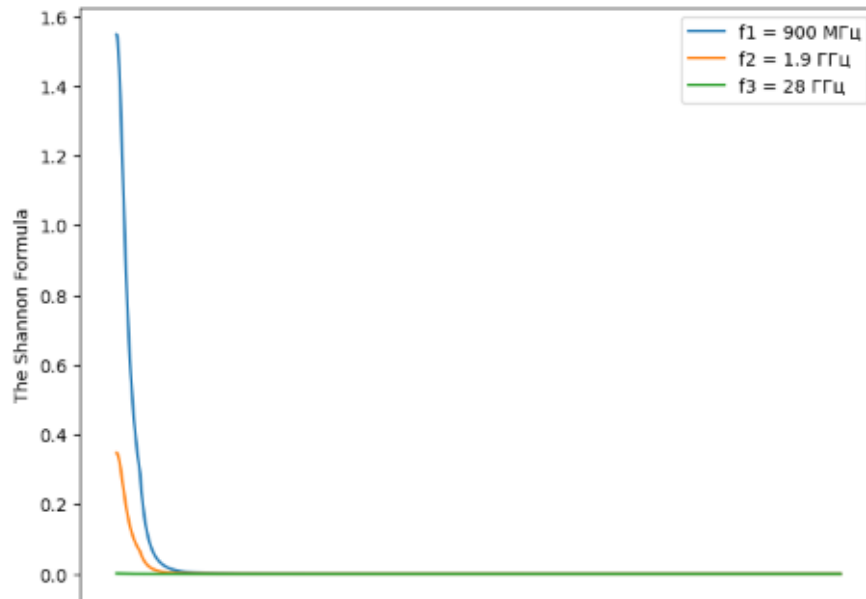

```

pwr_900, snr_900 = prd(G_r, G_t, P_t, d_inh, f[0], N_0, B, loss=inh_nlos, h_bs=h_bs_inh, h_ut=h_ut_inh)
pwr_1900, snr_1900 = prd(G_r, G_t, P_t, d_inh, f[1], N_0, B, loss=inh_nlos, h_bs=h_bs_inh, h_ut=h_ut_inh)
pwr_28000, snr_28000 = prd(G_r, G_t, P_t, d_inh, f[2], N_0, B, loss=inh_nlos, h_bs=h_bs_inh, h_ut=h_ut_inh)

shennon_900 = shennon(ln(snr_900), B)
shennon_1900 = shennon(ln(snr_1900), B)
shennon_28000 = shennon(ln(snr_28000), B)

plt.figure(figsize=(8,6))
plt.plot(d_inh, shennon_900, label='f1 = 900 МГц')
plt.plot(d_inh, shennon_1900, label='f2 = 1.9 ГГц')
plt.plot(d_inh, shennon_28000, label='f3 = 28 ГГц')
plt.xlabel('distance')
plt.ylabel('The Shannon Formula')
plt.legend()
plt.show()

```



Заключение.

Изучены модели потерь сигнала в различных условиях распространения радиоволн, таких как городская среда (Urban Macro, LOS/NLOS) и внутри зданий (Indoor, LOS/NLOS). Проведены анализы влияния различных параметров (мощности передатчика, усиления антенн, уровня шума, расстояния и частоты) на значения отношения сигнал/шум (SNR) и пропускной способности канала связи, рассчитанных по формуле Шеннона.

В ходе лабораторной работы были выполнены следующие задачи и получены выводы:

1. Анализ зависимости SNR от параметров системы

- SNR уменьшается с увеличением расстояния между передатчиком и приёмником.
- На частотах выше 28 ГГц наблюдается более резкое падение SNR при увеличении расстояния.

2. Влияние уровня шума и полосы частот:

- Увеличение уровня шума приводит к снижению пропускной способности канала связи.
- Широкая полоса частот (20 МГц) обеспечивает более высокую пропускную способность, но при этом более чувствительна к шуму.

3. Различие моделей потерь сигнала:

- Модель UMA (Urban Macro) демонстрирует меньшие потери сигнала при прямой видимости (LOS) по сравнению с отсутствием прямой видимости (NLOS).
- Модель Indoor показывает, что внутри зданий потери сигнала выше, особенно при отсутствии прямой видимости (NLOS).

4. Влияние мощности передатчика и усиления антенн:

- Увеличение мощности передатчика линейно улучшает SNR, что в свою очередь повышает пропускную способность канала.
- Усиление антенн также значительно влияет на качество связи, особенно на больших расстояниях.

Полученные результаты могут быть использованы при проектировании беспроводных сетей связи, таких как сети 4G/5G, для выбора оптимальных параметров системы.