

# РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

## ОТЧЕТ

### ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 2

дисциплина: Построение и анализ

моделей беспроводных сетей 5G/6G

Студент:

Кубасов Владимир Юрьевич

ст.б. 1132249516

Группа:

НФИмд-01-24

Преподаватель:

Молчанов Дмитрий Александрович

Самуйлов Андрей Константинович

МОСКВА

2024г.

### **Цель работы:**

Исследовать показатели шумов, ширины полос пропускания для различных моделей 3GPP стандарта

### **Выполнение работы**

#### **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?

### **Листинг программы:**

```
SPEED_OF_LIGHT = 299_792_458;
PREPARED_CONST = 20 * math.log10(4 * math.pi / SPEED_OF_LIGHT);
def FSPL(d, f):
    result = np.zeros(len(d));
    for i in range(len(d)):
        result[i] = PREPARED_CONST + 20 * math.log10(d[i] * f);
    return result;

Pt = 23;
G = 10;
B = 20e+6;
N = -174;
Pn = 10 * math.log10(B) + N;
```

```

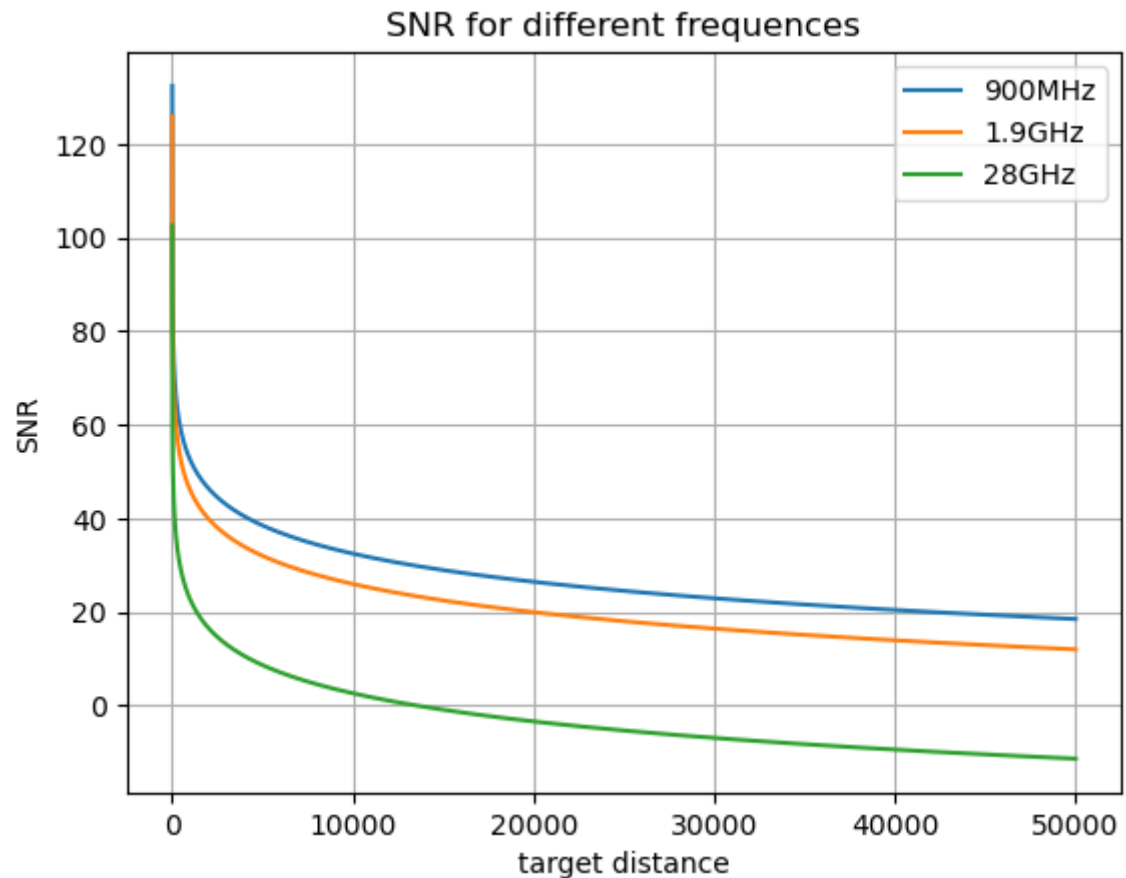
def SNR(fspl, gain = 10, Powt = 23, Pown = 10 * math.log10(B) + N):
    result = np.zeros(len(fspl));
    for i in range(len(fspl)):
        result[i] = Powt + gain * 2 - fspl[i];
        result[i] -= Pown;
    return result;

FREQ_1 = 900_000_000;
FREQ_2 = 1_900_000_000;
FREQ_3 = 28_000_000_000;
DISTANCE = np.linspace(0.1, 50000, 50000);
plt.plot(DISTANCE, SNR(FSPL(DISTANCE, FREQ_1)), label = "900MHz");
plt.plot(DISTANCE, SNR(FSPL(DISTANCE, FREQ_2)), label = "1.9GHz");
plt.plot(DISTANCE, SNR(FSPL(DISTANCE, FREQ_3)), label = "28GHz");
plt.xlabel("target distance");
plt.ylabel("SNR");
plt.legend();
plt.grid();
plt.title("SNR for different frequencies");

```

Так как первый пункт состоял больше из повторения лекции, переходим сразу ко второму. Помимо копиясты из прошлой лабораторной работы мы вводим показатель SNR — отношение сигнал-шум, который и будем исследовать. Так как в течение всей работы было удобнее использовать логарифмическую шкалу, то и SNR сразу вычислял значения в дБ.

Остальной код интереса не представляет, т.к. был написан в первой лабораторной работе.



На вопрос, что здесь можно определить как уровень уверенного приёма достаточно сложен. Возможно, ответ был бы очевиднее, если бы шкала SNR была рассмотрена в абсолютных значениях. Но, судя по значениям, найти такое число, где мощность сигнала, доходящая до приемника, больше или равна шуму, будет проблематично, ведь оно лежит вдали от исследуемых 50км. В любом случае, было предположено, что для качественного порога нам необходим показатель 20дБм, а поэтому введено ограничение и найдены предельные расстояния:

#### Листинг программы:

```
SNR_LIMIT = 20;

def findMaxFistance(snr_values, distances):
    for i in range(len(snr_values)):
        if snr_values[i] < SNR_THRESHOLD:
            return distances[i - 1] if i > 0 else distances[0];
    return distances[-1];
```

```

snr_900 = SNR(FSPL(DISTANCE, FREQ_1));
snr_1900 = SNR(FSPL(DISTANCE, FREQ_2));
snr_28G = SNR(FSPL(DISTANCE, FREQ_3));

max_dist_900 = find_max_distance(snr_900, DISTANCE);
max_dist_1900 = find_max_distance(snr_1900, DISTANCE);
max_dist_28G = find_max_distance(snr_28G, DISTANCE);

print(f"Максимальная дистанция уверенного приема для 900 МГц: {max_dist_900:.2f} м");
print(f"Максимальная дистанция уверенного приема для 1.9 ГГц: {max_dist_1900:.2f} м");
print(f"Максимальная дистанция уверенного приема для 28 ГГц: {max_dist_28G:.2f} м");

plt.plot(DISTANCE, snr_900, label="900 MHz");
plt.plot(DISTANCE, snr_1900, label="1.9 GHz");
plt.plot(DISTANCE, snr_28G, label="28 GHz");
plt.axhline(y=SNR_LIMIT, color='r', linestyle='--');
plt.xlabel("target distance");
plt.ylabel("SNR");
plt.legend();
plt.grid();
plt.title("SNR и максимальная дистанция уверенного приема");
plt.show();

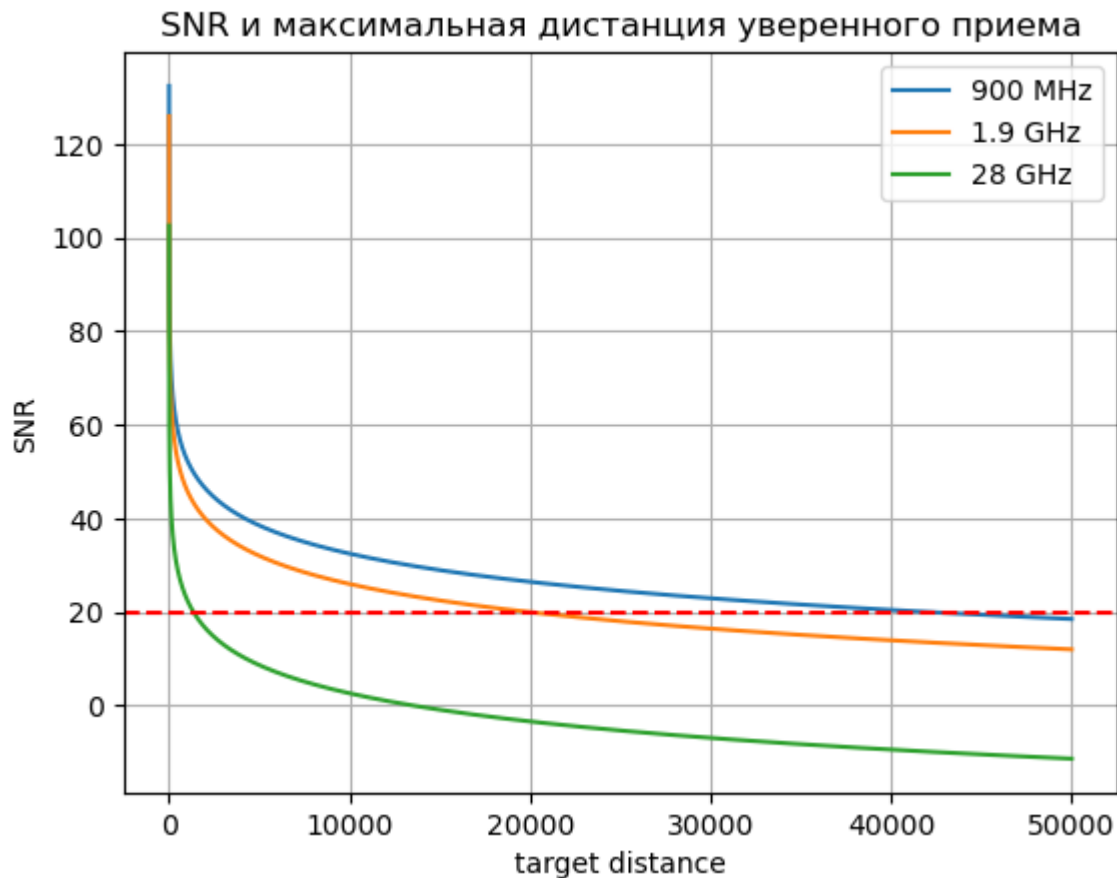
```

Вывод программы:

Максимальная дистанция уверенного приема для 900 МГц: 41960.86 м

Максимальная дистанция уверенного приема для 1.9 ГГц: 19876.46 м

Максимальная дистанция уверенного приема для 28 ГГц: 1348.12 м



Данные результаты по предельным расстояниям  $\pm$  сходятся с реальными, с учетом более щадящей FSPL-модели, но, с другой стороны, более жестких ограничений по пороговой мощности сигнала.

Если мы начнём увеличивать мощности, будь то антенна или усиление, мы будем повышать дистанцию уверенного приёма, согласно формуле. С обратной стороны, сейчас наши формулы не учитывают мощности, а следовательно, шум остаётся константным. Так что в такой вселенной с повышением мощностей на передатчиках или понижением уровня шума мы будем видеть один эффект.

Если у нас появляется интерференция от других вышек, но количественное её значение мы можем внести рядом с уровнем шумов. Причём интерференция не должна быть константой: она должна расти по мере удаления от базовой станции.

### Скорость по Шеннону

6. В условиях предыдущей задачи, постройте графики теоретической максимальной пропускной способности канала по теореме Шеннона от расстояния между приемником и передатчиком.

7. Почему скорость ненулевая даже для дистанций, на которых связь

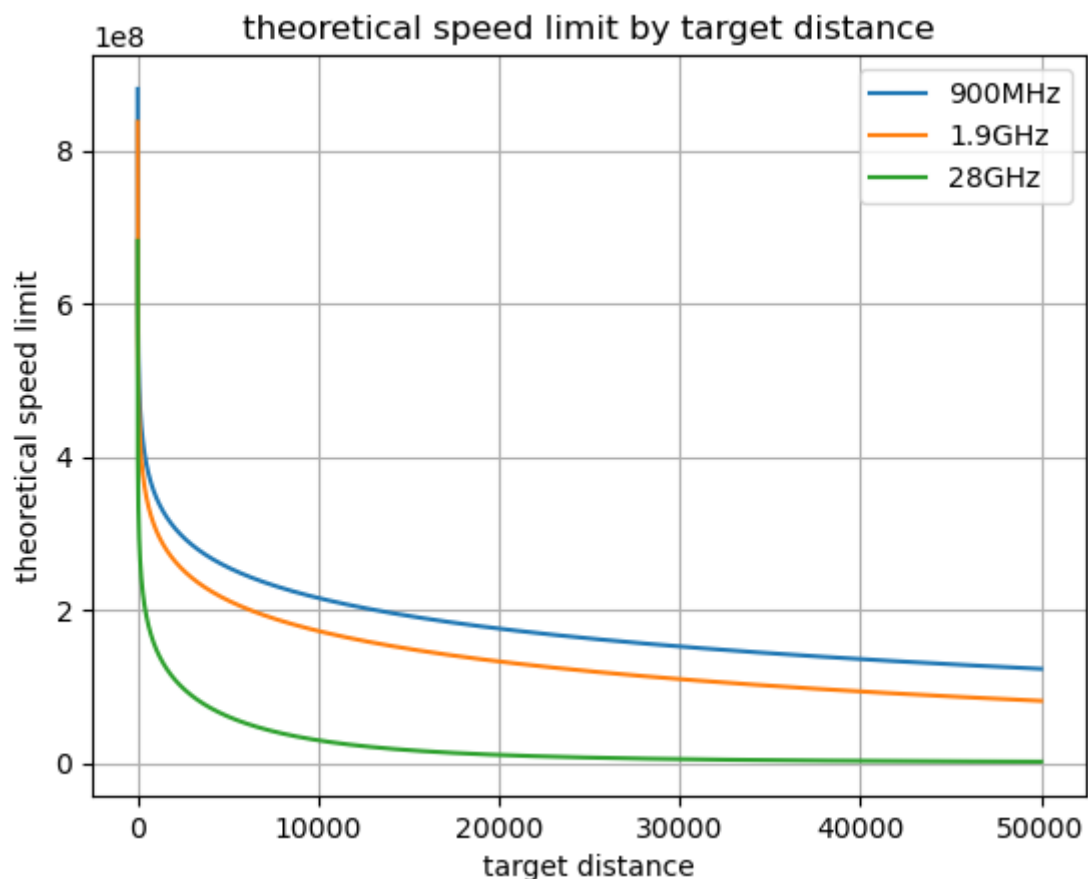
невозможна?

8. Зафиксировав дистанцию передачи, постройте графики зависимости пропускной способности от частоты, усиления антенны на передаче, мощности передатчика, и мощности шума.

9. За счет чего наиболее эффективно наращивать скорость беспроводных каналов связи?  $C = B \log_2 (1 + S N)$ , 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.

### **Листинг программы:**

```
I = 0;
def maxSpeedLimit(fspl):
    result = np.zeros(len(fspl));
    snr = SNR(fspl);
    for i in range(len(fspl)):
        result[i] = B * math.log2(1 + 10** (snr[i] / 10));
    return result;
plt.plot(DISTANCE, maxSpeedLimit(FSPL(DISTANCE, FREQ_1)), label = "900MHz");
plt.plot(DISTANCE, maxSpeedLimit(FSPL(DISTANCE, FREQ_2)), label = "1.9GHz");
plt.plot(DISTANCE, maxSpeedLimit(FSPL(DISTANCE, FREQ_3)), label = "28GHz");
plt.xlabel("target distance");
plt.ylabel("theoretical speed limit");
plt.legend();
plt.grid();
plt.title("theoretical speed limit by target distance");
```



Здесь будет лучше сразу привести график. По теореме Шеннона, это — предельная скорость. Де-факто это ограничение навряд ли достижимо на 100%. Ввиду того, что мы все вычисления ранее производили в логарифмической шкале, а пропускная способность не особо измеряется в дБ, переводим в абсолютные единицы и наблюдаем экспоненциальное убывание скоростей. Самый проблемный момент заключается в том, что большая часть пропускной способности уменьшается в пределах самой станции, а не на, например, пороговой дистанции.

Вопрос ненулевой скорости скорее относится к построению самой модели. Во время расчётов игнорируются другие факторы. Сейчас мы принимаем, что есть плоская земля без дождей и препятствий, которая держится на одной единственной базовой станции. К сожалению, данная утопия на нашу планету не распространяется, хотя эффект упадка пропускной способности к абсурдному минимуму мы можем наблюдать, например, в поезде на равнинной местности. Окажись наши устройства бесконечно чувствительными, мы бы действительно смогли принимать сигнал на очень больших расстояниях, однако, шумовые эффекты в один момент не позволили бы различить полезный сигнал, пусть он и старается побитово передать нам СМС.



### **Листинг программы:**

```
DISTANCE = 1000;

def parametrizedMaxSpeedLimit(frequency, gain, transmitterPower, noisePower):
    fspl = FSPL(np.array([DISTANCE]), frequency);
    snr = SNR(fspl, gain, transmitterPower, noisePower);
    return B * math.log2(1 + 10**(snr[0] / 10));

gains = np.linspace(0, 50, 100);
speeds_gain = [parametrizedMaxSpeedLimit(FREQ_2, g, Pt, Pn) for g in gains];

plt.plot(gains, speeds_gain, label="Antenna Gain vs Capacity");
plt.xlabel("Antenna Gain (dB)");
plt.ylabel("Channel Capacity (bps)");
plt.legend();
plt.grid();
plt.title("Channel Capacity vs Antenna Gain");
plt.show();

frequencies = np.linspace(100_000_000, 30_000_000_000, 100);
speeds_freq = [parametrizedMaxSpeedLimit(f, G, Pt, Pn) for f in frequencies];

plt.plot(frequencies, speeds_freq, label="Frequency vs Capacity");
plt.xlabel("Frequency (Hz)");
plt.ylabel("Channel Capacity (bps)");
plt.legend();
plt.grid();
plt.title("Channel Capacity vs Frequency");
plt.show();

transmitters = np.linspace(0, 50, 100);
speeds_power = [parametrizedMaxSpeedLimit(FREQ_2, G, p, Pn) for p in transmitters];

plt.plot(transmitters, speeds_power, label="Transmitter Power vs Capacity");
plt.xlabel("Transmitter Power (dBm)");
plt.ylabel("Channel Capacity (bps)");
```

```

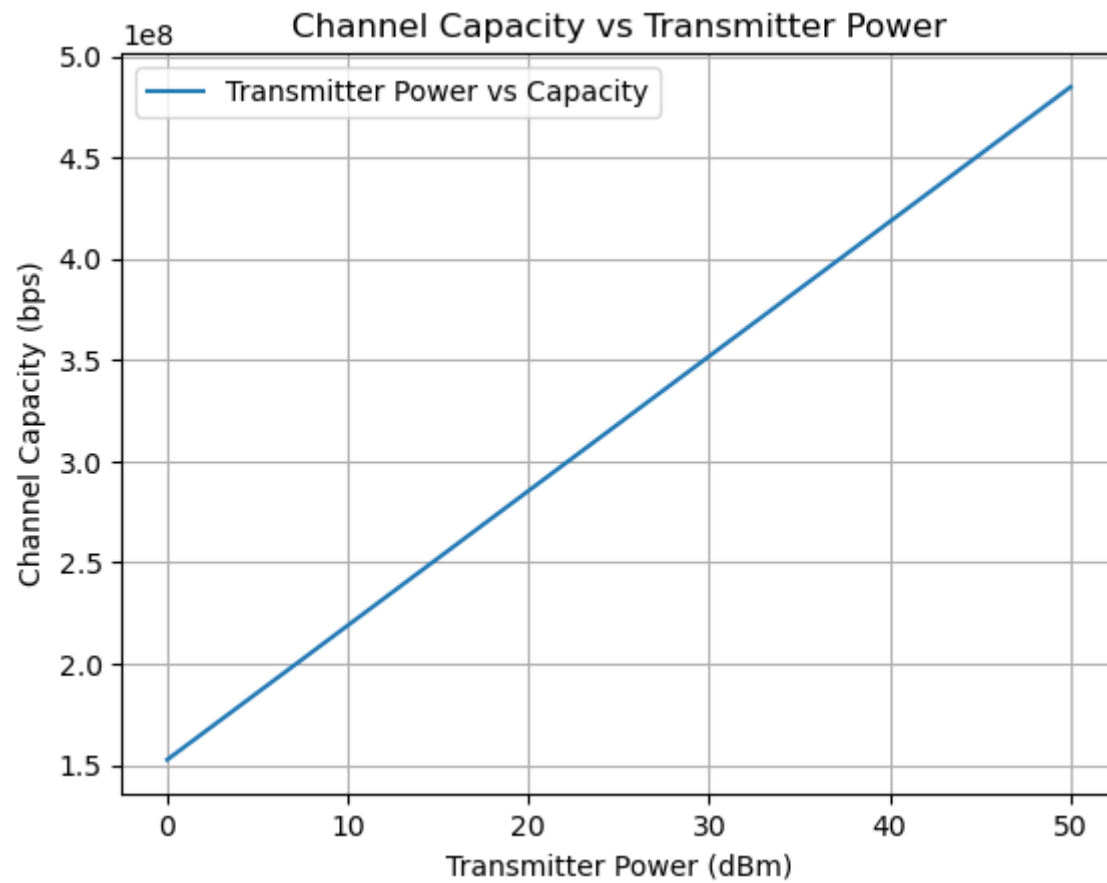
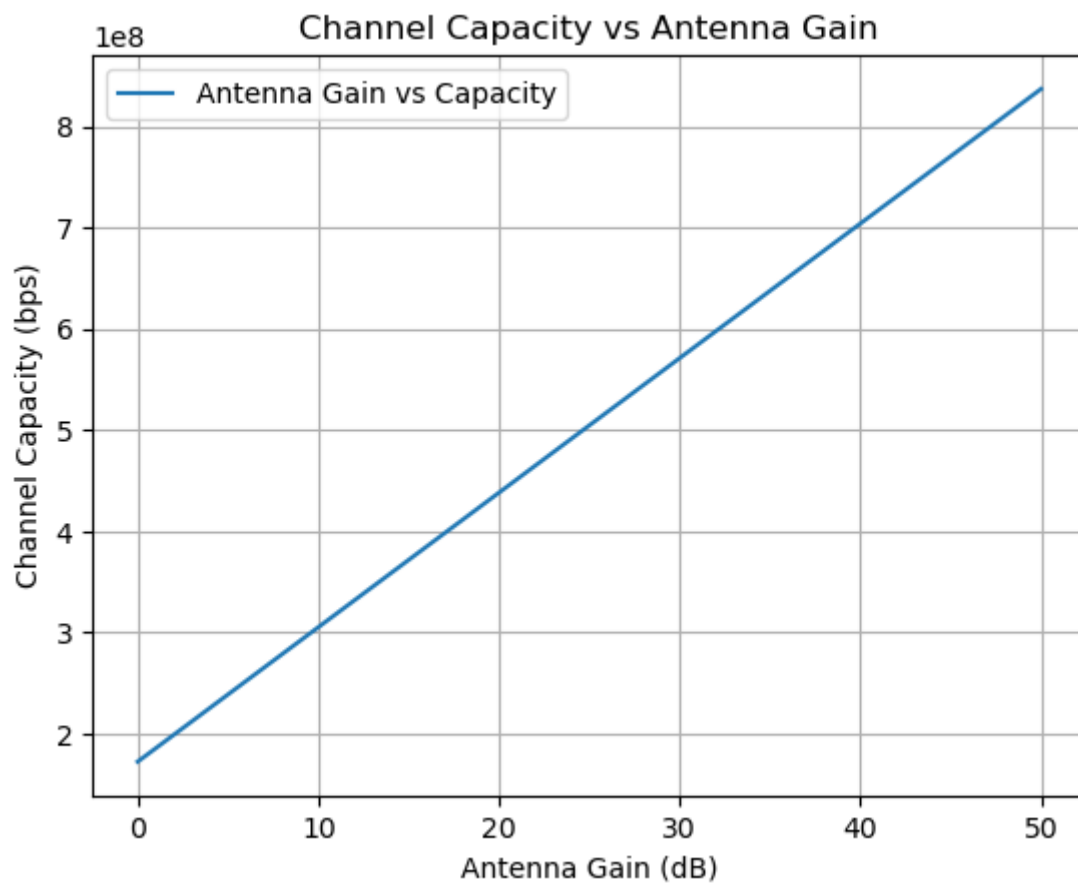
plt.legend();
plt.grid();
plt.title("Channel Capacity vs Transmitter Power");
plt.show();

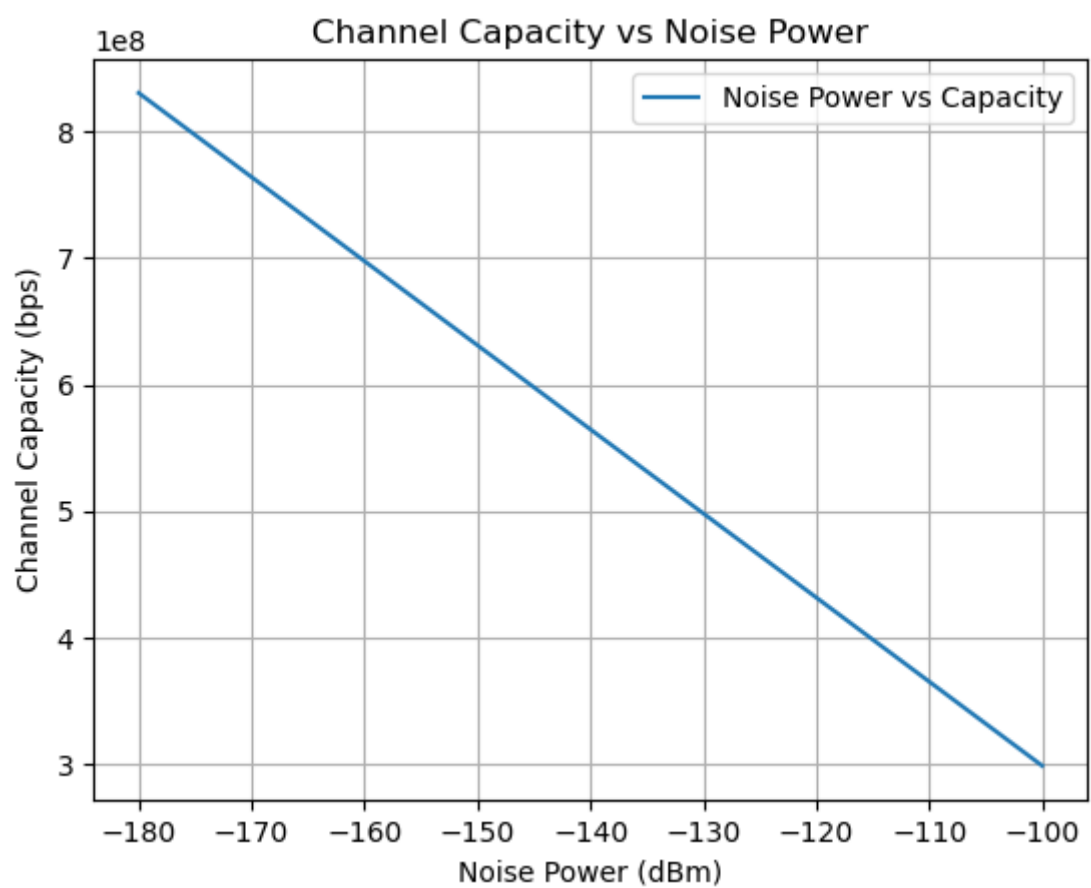
noises = np.linspace(-180, -100, 100);
speeds_noise = [parametrizedMaxSpeedLimit(FREQ_2, G, Pt, n) for n in noises];

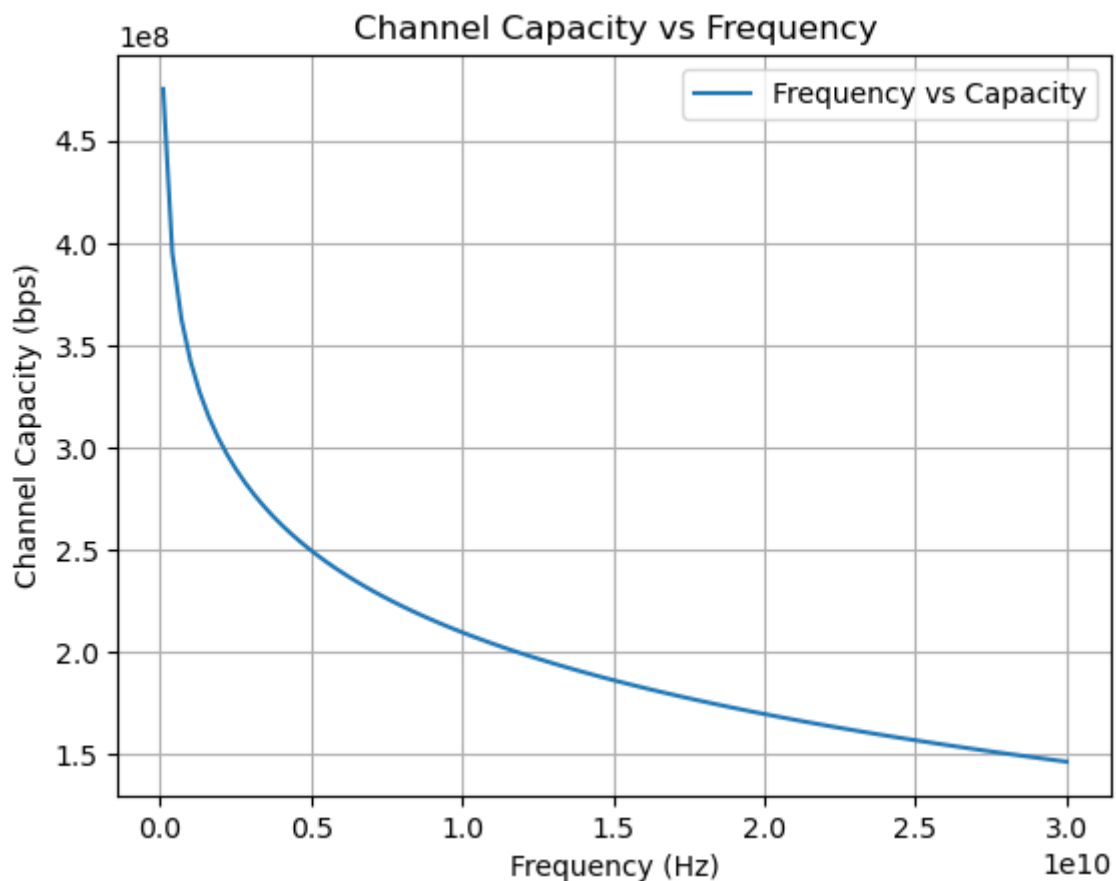
plt.plot(noises, speeds_noise, label="Noise Power vs Capacity");
plt.xlabel("Noise Power (dBm)");
plt.ylabel("Channel Capacity (bps)");
plt.legend();
plt.grid();
plt.title("Channel Capacity vs Noise Power");
plt.show();

```

В данном пункте мы исследуем пропускную способность с точки зрения варьирования прочих характеристик (изменения мощностей и т. д.). Выводы данных программ:







Расстояние от базовой станции в 1км было выбрано для наглядности.

В общем случае, мы получаем ожидаемые влияния характеристик на пропускную способность.

Вопрос, который поднимался не раз на лекциях. Как можно повысить пропускную способность согласно Шеннону? Увеличением ширины полосы пропускания. Приросты мощностей, согласно моделям, дают нам визуальный прирост, однако так же дадут повышение паразитных эффектов, стоимости связи и пр. неприятных моментов.

### 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?

**Листинг программы:**

```

def UMaLoS(d, f):
    if not isinstance(d, np.ndarray):
        d = [d]
    result = np.zeros(len(d));
    for i in range(len(d)):
        result[i] = 28 + 22 * math.log10(d[i]) + 20 * math.log10(f);
    return result;

BASE_ST_HEIGHT = 25;

def UManLoS(d, f):
    result = np.zeros(len(d));
    for i in range(len(d)):
        result[i] = max(UMaLoS(d[i], f), 13.54 + 39.08 * math.log10(d[i]) + 20 *
math.log10(f) - 0.6 * (BASE_ST_HEIGHT - 1.5));
    return result;

def InHOfficeLoS(d, f):
    result = np.zeros(len(d));
    for i in range(len(d)):
        result[i] = 32.4 + 17.3 * math.log10(d[i]) + 20 * math.log10(f);
    return result;

def InHOfficeNLoS(d, f):
    result = np.zeros(len(d));
    for i in range(len(d)):
        result[i] = 38.3 + 24.9 * math.log10(d[i]) + 20 * math.log10(f);
    return result;

DISTANCE = np.linspace(0.1, 10000, 10000);

plt.plot(DISTANCE, maxSpeedLimit(UMaLoS(DISTANCE, FREQ_2)), label = "Городская
среда");

plt.plot(DISTANCE, maxSpeedLimit(UManLoS(DISTANCE, FREQ_2)), label = "Городская
среда без п.в.");

plt.plot(DISTANCE, maxSpeedLimit(InHOfficeLoS(DISTANCE, FREQ_2)), label =
"Офисные помещения");

plt.plot(DISTANCE, maxSpeedLimit(InHOfficeNLoS(DISTANCE, FREQ_2)), label =
"Офисные помещения без п.в.");

plt.xlabel("target distance");

plt.ylabel("theoretical speed limit");

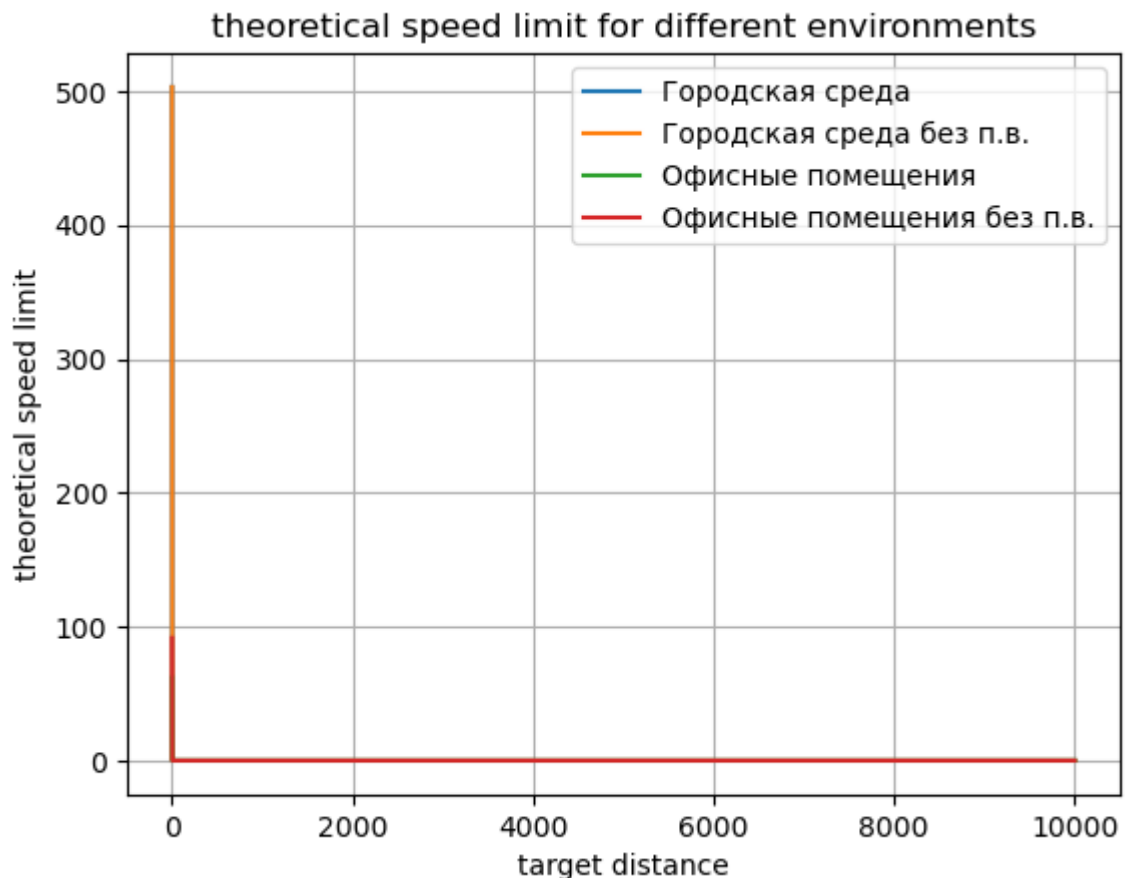
plt.legend();

plt.grid();

```

```
plt.title("theoretical speed limit for different environments");
```

В данном пункте я ожидал достаточно сильного понижения пропускной способности, по сравнению с FSPL, однако результаты расчётов оставили меня в рассуждениях правильности вычислений.



Если попробовать подставить вместо прочих моделей из 3GPP FSPL, то результат высчитывается прекрасно. Возможно это связано с устройством языка, а также вычислений достаточно маленьких чисел в логарифме, на которых регулярно сыпятся все вычисления. Ожидалось тут получить более резко-уходящую вниз кривую, нежели FSPL.

Согласно предыдущей лабораторной работе, из предложенной четверки меньше всего потери у офисной среды с прямой видимостью, или InH-Office LoS, поэтому ответ, в какой среде достижимая скорость выше — именно такой (ввиду меньших потерь).

**Выводы:**

Исследовали такой показатель как соотношение полезного сигнала к шуму, рассчитали максимальные теоретические пропускные способности для различных моделей стандарта 3GPP в зависимости от различных, входящих в них, параметров.