

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 6

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:

Исследование среднего показателя интерференции на основании ранее полученных формул и результатов.

Выполнение работы

1. Получите графики зависимости среднего значения интерференции от параметров (направленности антенн α , плотностей интерферирующих устройств λ_I) на случайно выбранном АУ используя формулу Кэмбелла при предположениях, что поле blockers – Пуассоновское с некоторой плотностью λ_B , используя модель FSPL, предположив дополнительно излучаемую мощность антенны БС – 23 дБм, усиления на передаче и приеме – 10 дБ, радиус области интерференции – 1 км. Проанализируйте полученные кривые. Что происходит с интерференцией при изменении параметров и почему?
2. Предположите, что blockers отсутствуют ($\lambda_B=0$). Получите результаты и сравните с пунктом 1. Что происходит с интерференцией? Каково влияние blockers на интерференцию?
3. Рассмотрите системы с всенаправленными антеннами. Посчитайте среднее значение интерференции и поясните полученные результаты.

Для начала получим необходимые формулы:

Найдем коэффициент А как:

$$A = \frac{P_t G_t G(\alpha)}{(4\pi)^2}$$

Тогда полезная мощность:

$$P_R = A r^{-2} e^{-K r}$$

Вероятность экспозиции:

$$p_c = \left(\frac{\alpha}{2\pi}\right)^2$$

Вероятность прямой видимости будем считать при условии, что высота станции равна высоте блокировки для простоты:

$$p_l = e^{-2\lambda_B R_B(R_B+r)}$$

Тогда итоговая средняя интерференция:

$$E(I) = \int_{R_B}^R P_R p_c p_l 2\lambda_I \pi r dr$$

Если подставить все функции и собрать итоговую формулу матожидания, то мы увидим, что аналитически интеграл не берётся, а, следовательно, придется использовать численные методы

Далее опишем необходимые функции из приведённого выше фрагмента:

Листинг программы:

```

linear = lambda x: 10 ** (x / 10);
alpha = 0.4;
pc = lambda alpha : (alpha / 2 / math.pi) ** 2;
k = 0.1;
K = 0.3;
A = lambda alpha : linear(2 / (1 - math.cos(alpha / 2) + k * (1 + math.cos(alpha / 2)))) * linear(10) * linear(23) / (4 * math.pi) ** 2;
Pr = lambda r : r ** (-2) * math.exp(-K * r);
lmbdB = 0.2;
Rb = 1.5;
pl = lambda r : math.exp(-2 * lmbdB * Rb * (Rb + r));
R = 1000;
underIntEI = lambda r: Pr(r) * pl(r) * 2 * math.pi * r;

```

По формулам видно, что мы считаем случайной величиной только расстояние, остальные показатели считаем константными. В таком случае всё необходимое было вынесено из-под интеграла для дальнейшей вариации, итоговый результат формировался далее.

Листинг программы:

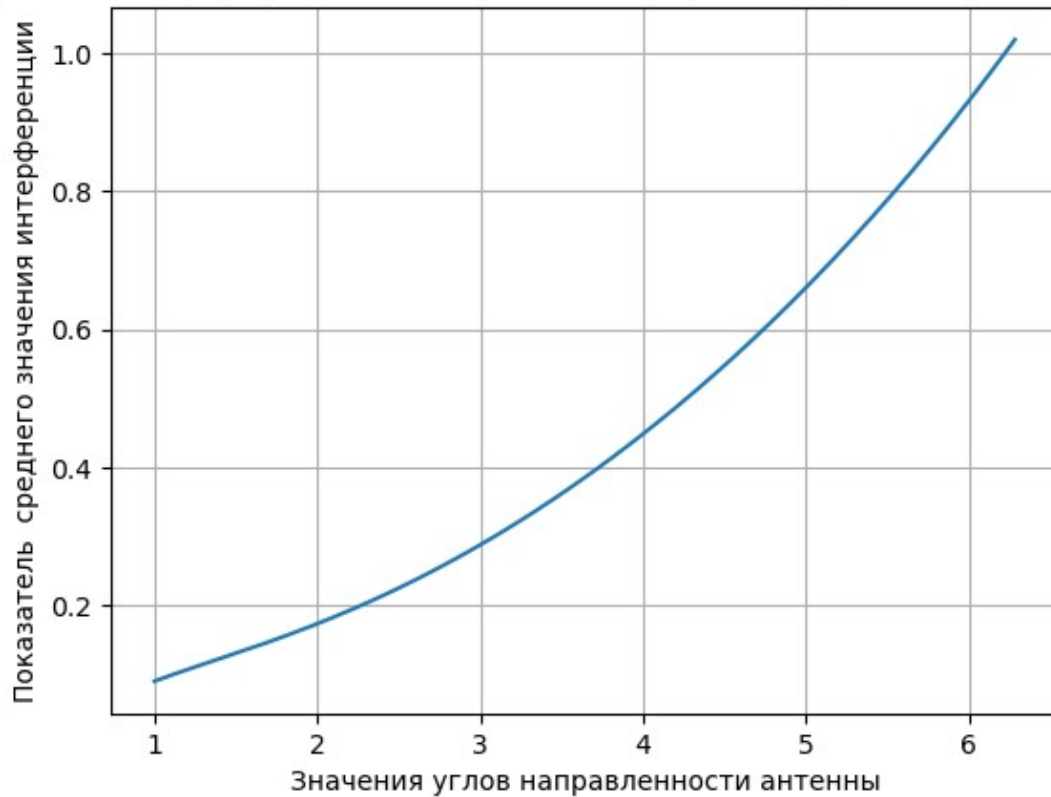
```

POINTS = 1000;
def getByAngle():
    lmbdI = 0.2;
    angles = np.linspace(1, 2 * math.pi, POINTS);
    result = np.zeros(POINTS);
    for i in range(POINTS):
        alpha = angles[i];
        (result[i], scale) = scipy.integrate.quad(underIntEI, Rb, R);
        result[i] *= pc(alpha) * lmbdI * A(alpha);
    plt.plot(angles, result);
    plt.xlabel("Значения углов направленности антенны");
    plt.ylabel("Показатель среднего значения интерференции");
    plt.title("Средняя интерференция в зависимости от угла направленности антенны");
    plt.grid();

```

```
getByAngle();  
plt.show();
```

Средняя интерференция в зависимости от угла направленности антенны



В данном пункте я пытался выводить график в зависимости сначала константного усиления (то есть при $G = \text{const}$), затем для первой модели усиления (без потерь на боковые лепестки, однако именно учёт потерь на боковые лепестки дал приемлемую картину).

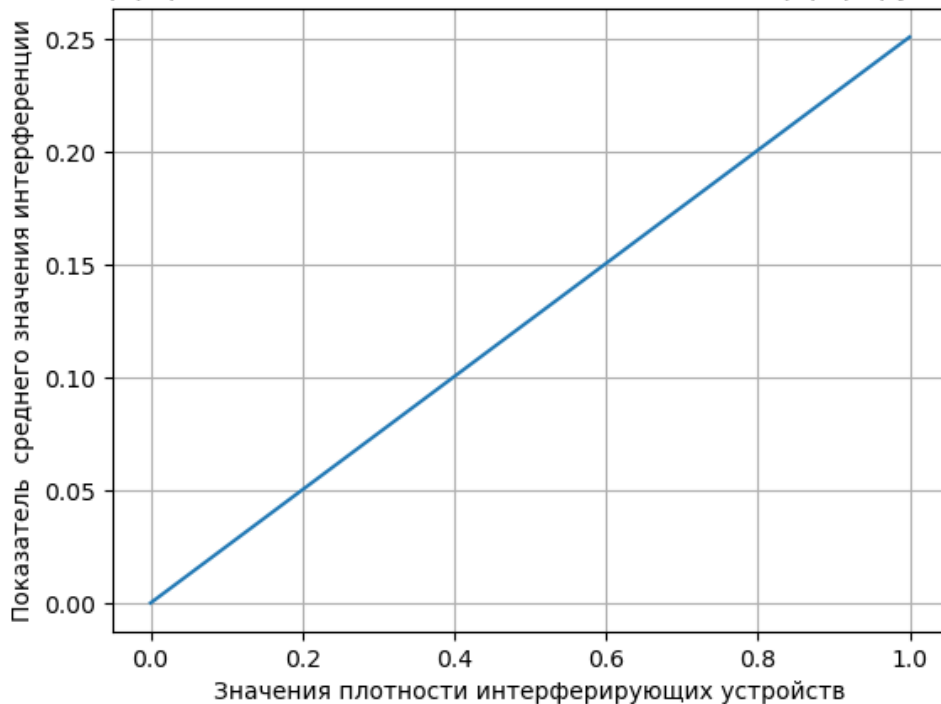
Далее получим аналогичный график для изменения плотности интерферирующих устройств:

Листинг программы:

```
def getByLambdaI():  
    alpha = 0.6;  
    result = np.zeros(PPOINTS);  
    lmbdsI = np.linspace(0, 1, PPOINTS);  
    for i in range(PPOINTS):  
        (result[i], scale) = scipy.integrate.quad(underIntEI, Rb, R);  
        result[i] *= pc(alpha) * lmbdsI[i] * A(alpha);
```

```
plt.plot(lmbdsI, result);  
plt.xlabel("Значения плотности интерферирующих устройств");  
plt.ylabel("Показатель среднего значения интерференции");  
plt.title("Средняя интерференция в зависимости от плотности интерферирующих устройств");  
plt.grid();  
getByLambdaI();  
plt.show();
```

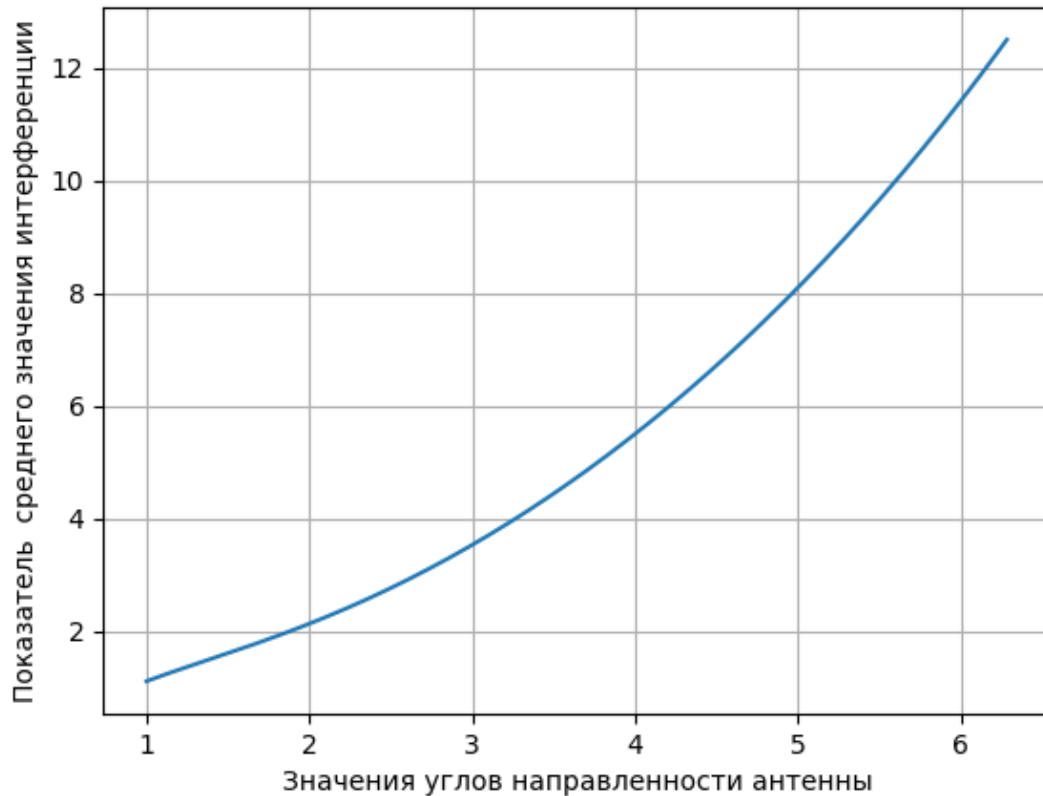
Средняя интерференция в зависимости от плотности интерферирующих устройств



Так как плотность интерферирующих устройств выносится из-под интеграла и является линейным значением, то логично здесь было ожидать линейное увеличение мат ожидания.

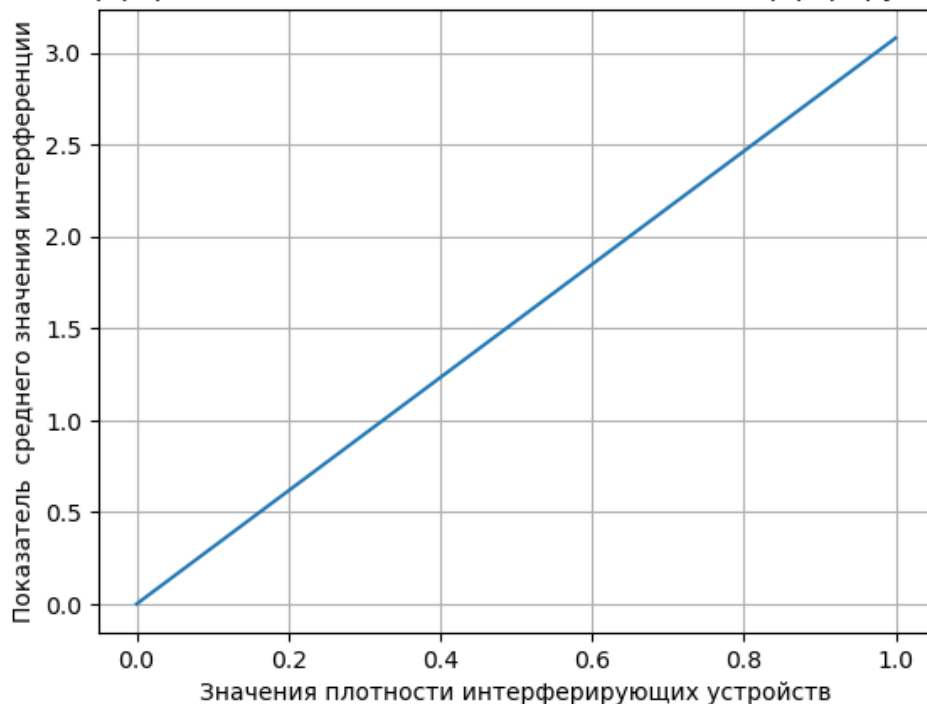
Далее исключаем блокировки из данных вычислений, т. к. код составлялся с помощью встроенных функций «Копировать» + «Вставить», то листинг не привожу.

Средняя интерференция в зависимости от угла направленности антенны



Сама функция не изменилась, однако резко выросли значения интерференции. Обосновать это можно достаточно просто. Ввиду того, что блокировки аналогично воздействовали как на сигнал, так и на помехи, то исключение блокировок привело к резкому росту интерференции на приёмнике. Далее для плотности интерферирующих устройств:

Средняя интерференция в зависимости от плотности интерферирующих устройств



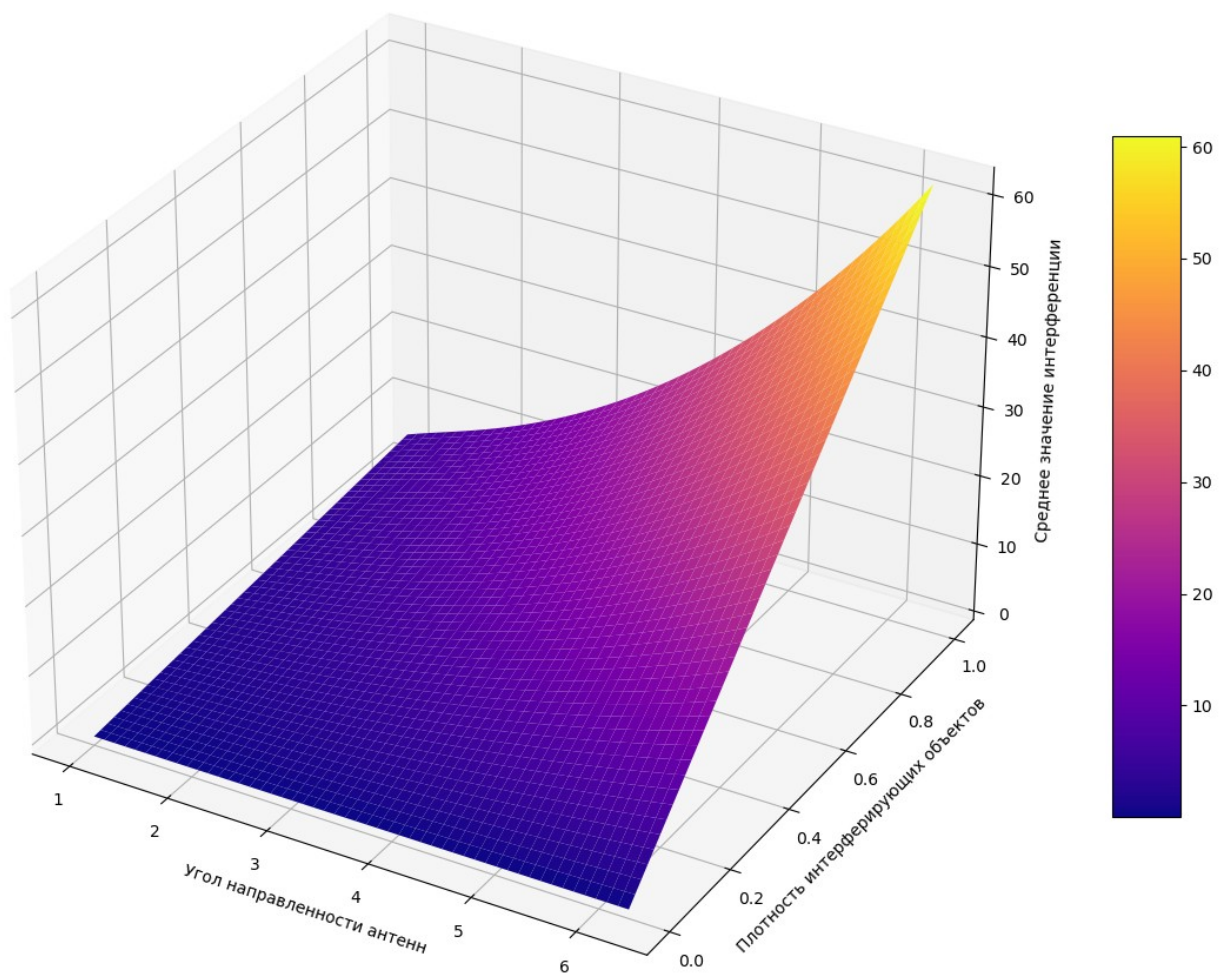
Аналогично предыдущему графику. Произошёл резкий рост значений ввиду исключения блокировок.

Далее для большей наглядности был введён 3D график для взаимного воздействия как изменений направленности угла, так и плотности интерферирующих устройств.

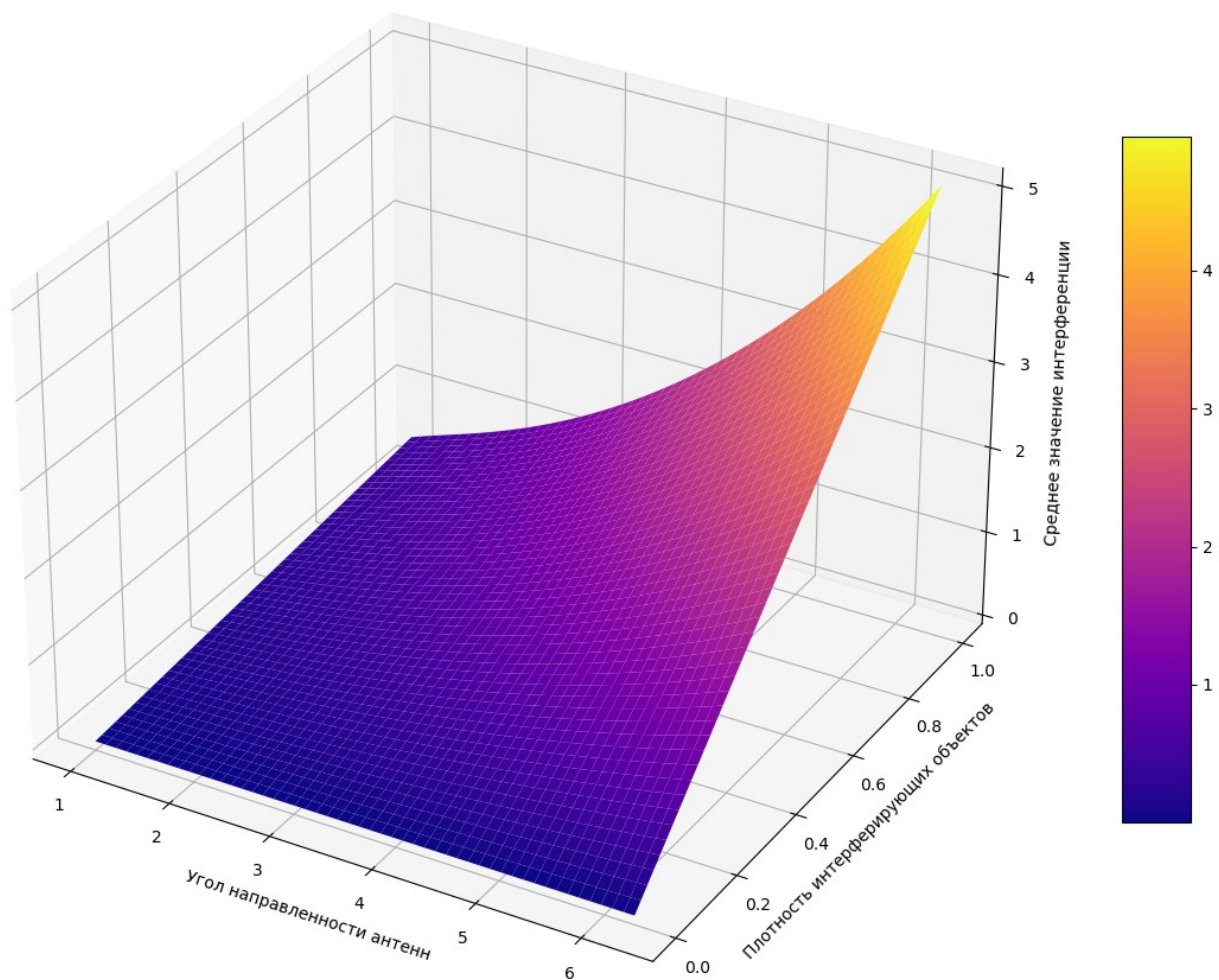
Листинг программы:

```
angles = np.linspace(1, 2 * math.pi, POINTS);
lmbdsI = np.linspace(0, 1, POINTS);
X, Y = np.meshgrid(angles, lmbdsI);
result = np.zeros((POINTS, POINTS));
(temp, scale) = scipy.integrate.quad(underIntEI, Rb, R);
for i in range(POINTS):
    for j in range(POINTS):
        result[i][j] = temp * pc(X[i][j]) * Y[i][j] * A(X[i][j]);
fig = plt.figure(figsize=(15, 15));
ax = fig.add_subplot(111, projection='3d');
scatter = ax.plot_surface(X, Y, result, cmap='plasma');
cbar = plt.colorbar(scatter, ax=ax, shrink=0.5, aspect=10);
ax.set_xlabel('Угол направленности антенн');
ax.set_ylabel('Плотность интерферирующих объектов');
ax.set_zlabel('Среднее значение интерференции');
```

```
plt.show();
```



Здесь уже более точно видно воздействие обеих изменяемых величин на показатель интерференции. Далее, попробуем вернуть блокирующие объекты и отследить изменение экстремума средней интерференции (по коду снова копипаста, поэтому, не прикладываю):



Можно заметить, что отсутствие блокировок уменьшило мат ожидание интерференции почти в 12 раз, что достаточно ощутимо. В отсутствие блокирующих объектов, например, при отсутствии различной инфраструктуры, связь была бы ощутимо проблемнее. Так что в данном случае, блокировки играют нам на руку.

Выводы:

По данной лабораторной работе можно сделать вывод, что, во-первых, показатель интерференции сложная функция от других различных моделей, а следовательно, она так же будет изменяться при изменении моделей распространения сигнала в среде и изменении группы понижающих коэффициентов.

Во-вторых, достаточное количество блокирующих объектов может помочь в обеспечении более качественных услуг связи ввиду уменьшения ими интерференции.