



**UNIVERSIDADE FEDERAL DE UBERLÂNDIA**  
**FACULDADE DE ENGENHARIA ELÉTRICA**  
**ENGENHARIA ELETRÔNICA E DE TELECOMUNICAÇÕES**

---

**JOSÉ GABRIEL RIBEIRO FAGUNDES**

**Análise dos impactos da difração e do posicionamento das EBs no desempenho de redes 5G utilizando o simulador Vienna 5G System Level.**

UBERLÂNDIA

2024

**José Gabriel Ribeiro Fagundes**

**Análise dos impactos da difração e do posicionamento das EBs no desempenho de redes 5G utilizando o simulador Vienna 5G System Level.**

Trabalho de conclusão de curso apresentado ao Curso de graduação em Engenharia Eletrônica e de telecomunicações da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Engenheiro Eletrônico e de telecomunicações grau bacharelado.

Orientador: Prof. Dr. Éderson Rosa da Silva.

Uberlândia

2024

*Gostaria de dedicar esse trabalho primeiramente a DEUS por estar sempre me fortalecendo e me guiando com sua imensa sabedoria, mesmo com todas as limitações que possuo. Agradeço a ele por me guiar e sempre me mostrar o caminho até mesmo quando as situações pareciam não ter solução. Devo a ele tudo de bom que há em mim, e com todo esforço que faço, ainda não consigo retribuí-lo por toda sua bondade e amor comigo. Dedico também à meus pais, a minha irmã e a todos os meus irmãos em Cristo.*

## **Agradecimentos**

Agradeço primeiramente a DEUS por tornar tudo isso possível e por me dar forças em todos os momentos.

Aos meus pais, José Marcos e Rosimeire, por todo apoio, carinho, inspiração e toda a sabedoria dos seus conselhos durante todo o tempo, e por nunca medirem esforços para que eu pudesse estudar e realizar os meus sonhos. Sem eles, eu não estaria aonde estou hoje.

A Stéfany, minha irmã, por todo apoio, carinho, companhia e conhecimento compartilhado, sempre me incentivando a aprender coisas novas e me ajudando a ver os acontecimentos de uma forma diferente.

Ao meu orientador, Prof. Dr. Éderson Rosa da Silva, por toda a dedicação, por seus ensinamentos e pelo tempo dedicado a tirar minhas dúvidas, além de seus excelentes conselhos que levarei para a vida.

Aos professores com quem tive a oportunidade de aprender, por todo o conhecimento transmitido e pela qualidade e dedicação com que fizeram isso, principalmente aos professores do curso de Engenharia Eletrônica e de Telecomunicações, com quem passei a maior parte da graduação.

Aos meus amigos e colegas por todos os momentos em que estivemos juntos.

A minha família, por todos os grandes momentos que passamos juntos durante a graduação, e por todo incentivo e toda ajuda com os assuntos que vi no decorrer do curso.

## Resumo

Com a crescente migração dos sistemas de 4G para 5G, as operadoras passam por alguns desafios relacionados a essa nova tecnologia. Em sistemas celulares consolidados, é preciso analisar as células existentes e verificar se a posição de cada estação-base é adequada para a implantação dessa tecnologia (5G). O propósito deste trabalho é analisar os impactos da difração e do posicionamento da estação-base em uma rede móvel 5G fazendo uso do simulador Vienna 5G System Level, que não considera a perda por difração. Os resultados indicam uma melhora na vazão quando a posição da estação-base é otimizada, justificada pelo fato da perda por difração ser quase insignificante. Por outro lado, resultados imprecisos são observados quando a posição da estação-base não é otimizada, ao não se considerar a perda por difração.

**Palavras-chave:** Difração, Otimização, Redes 5G, Vienna 5G System Level Simulator.

## **Abstract**

With the increasing migration of systems from 4G to 5G, operators are experiencing some challenges related to this new technology. In consolidated cellular systems, it is necessary to analyze the existing cells and check whether the position of each base station is suitable for the implementation of this technology (5G). The purpose of this work is to analyze the impacts of diffraction and base station positioning in a 5G mobile network using the Vienna 5G System Level simulator, which does not consider diffraction loss. The results indicate an improvement in throughput when the position of the base station is optimized, justified by the fact that the diffraction loss is almost insignificant. On the other hand, inaccurate results are observed when the position of the base station is not optimized, by not considering diffraction loss.

**Key-words:** 5G Networks, Diffraction, Optimization, Vienna 5G System Level Simulator.

## Sumário

<b>Capítulo 1 - Introdução</b> .....	<b>14</b>
1.1. Motivação .....	15
1.2. Objetivos .....	16
1.3. Organização do trabalho .....	16
<b>Capítulo 2 – Fundamentação Teórica</b> .....	<b>17</b>
2.1. Propagação de rádio móvel .....	17
2.1.1. Introdução à propagação de ondas .....	17
2.1.2. Modelos e perdas básicas de propagação .....	19
2.1.3. Perda por difração .....	22
2.2. Comunicações digitais .....	26
2.2.1. Introdução .....	26
2.2.2. Arquitetura típica de um receptor digital .....	27
2.2.3. Impactos da relação sinal-ruído na recepção de um sinal .....	28
2.3. Visão geral do 5G .....	29
2.4. Simulador Vienna 5G System Level.....	30
2.4.1. Introdução .....	30
2.4.2. Simulação no simulador Vienna 5G SL.....	31
2.4.3. Modelos utilizados do simulador .....	33
2.5. Considerações Finais .....	33
<b>Capítulo 3 - Metodologia</b> .....	<b>35</b>
3.1. Introdução .....	35
3.2. Software para cálculo de perda .....	38
3.3. Software para otimizar a posição da estação-base .....	39
3.4. Cenários de simulação .....	41
3.5. Considerações finais .....	47
<b>Capítulo 4 - Resultados</b> .....	<b>49</b>
4.1. Introdução .....	49
4.2. Resultados dos cálculos das perdas por difração .....	49
4.3. Resultados de simulação .....	58
4.4. Considerações finais .....	65
<b>Capítulo 5 - Conclusão</b> .....	<b>66</b>

<b>Referências .....</b>	<b>68</b>
<b>Apêndice A .....</b>	<b>69</b>
<b>Apêndice B .....</b>	<b>76</b>

## Lista de figuras

<b>Figura 2.1. Representação da difração .....</b>	<b>18</b>
<b>Figura 2.2. Geometria para o modelo de dois raios.....</b>	<b>20</b>
<b>Figura 2.3. Geometria da difração do gume de faca .....</b>	<b>23</b>
<b>Figura 2.4. Ganho da difração do tipo gume de faca em função do parâmetro <math>v</math> .....</b>	<b>26</b>
<b>Figura 2.5. Página do openStreetMap .....</b>	<b>30</b>
<b>Figura 2.6. Trecho do código responsável por definir a região, no cenário openStreetMap do simulador .....</b>	<b>31</b>
<b>Figura 3.1. Página principal do Google Cloud Platform .....</b>	<b>37</b>
<b>Figura 3.2. Ilustração do software de otimização da posição da EB .....</b>	<b>40</b>
<b>Figura 3.3. Cenário A. Posição da EB não otimizada .....</b>	<b>45</b>
<b>Figura 3.4. Cenário A. Posição da EB otimizada.....</b>	<b>46</b>
<b>Figura 3.5. Cenário B. Posição da EB não otimizada.....</b>	<b>46</b>
<b>Figura 3.6. Cenário B. Posição da EB otimizada.....</b>	<b>47</b>
<b>Figura 4.1. Perfil do terreno entre U1 e a EB no cenário A. Posição da EB não otimizada .....</b>	<b>52</b>
<b>Figura 4.2. Perfil do terreno entre U2 e a EB no cenário A. Posição da EB não otimizada .....</b>	<b>53</b>
<b>Figura 4.3. Perfil do terreno entre U3 e a EB no cenário A. Posição da EB não otimizada .....</b>	<b>53</b>
<b>Figura 4.4. Perfil do terreno entre U1 e a EB no cenário A. Posição da EB otimizada ...</b>	<b>54</b>
<b>Figura 4.5. Perfil do terreno entre U2 e a EB no cenário A. Posição da EB otimizada ...</b>	<b>54</b>
<b>Figura 4.6. Perfil do terreno entre U3 e a EB no cenário A. Posição da EB otimizada ...</b>	<b>55</b>
<b>Figura 4.7. Perfil do terreno entre U1 e a EB no cenário B. Posição da EB não otimizada .....</b>	<b>55</b>
<b>Figura 4.8. Perfil do terreno entre U2 e a EB no cenário B. Posição da EB não otimizada .....</b>	<b>56</b>
<b>Figura 4.9. Perfil do terreno entre U3 e a EB no cenário B. Posição da EB não otimizada .....</b>	<b>56</b>
<b>Figura 4.10. Perfil do terreno entre U1 e a EB no cenário B. Posição da EB otimizada .</b>	<b>57</b>
<b>Figura 4.11. Perfil do terreno entre U2 e a EB no cenário B. Posição da EB otimizada .</b>	<b>57</b>

<b>Figura 4.12. Perfil do terreno entre U3 e a EB no cenário B. Posição da EB otimizada .</b>	<b>58</b>
<b>Figura 4.13. Resultado: Vazão para o cenário A. Posição da EB não otimizada .....</b>	<b>59</b>
<b>Figura 4.14. Resultado: SINR para o cenário A. Posição da EB não otimizada .....</b>	<b>60</b>
<b>Figura 4.15. Resultado: Vazão para o cenário A. Posição da EB otimizada.....</b>	<b>60</b>
<b>Figura 4.16. Resultado: SINR para o cenário A. Posição da EB otimizada.....</b>	<b>61</b>
<b>Figura 4.17. Resultado: Vazão para o cenário B. Posição da EB não otimizada.....</b>	<b>62</b>
<b>Figura 4.18. Resultado: SINR para o cenário B. Posição da EB não otimizada.....</b>	<b>63</b>
<b>Figura 4.19. Resultado: Vazão para o cenário B. Posição da EB otimizada.....</b>	<b>63</b>
<b>Figura 4.20. Resultado: SINR para o cenário B. Posição da EB otimizada.....</b>	<b>64</b>

## Lista de tabelas

<b>Tabela 2.1. Modelo de canal ITU Pedestrian A .....</b>	<b>33</b>
<b>Tabela 3.1. Parâmetros principais de simulação do Vienna 5G SL.....</b>	<b>42</b>
<b>Tabela 3.2. Parâmetros principais dos usuários no cenário de simulação do Vienna 5G SL .....</b>	<b>42</b>
<b>Tabela 3.3. Coordenadas dos usuários e da EB para o cenário A. Posição da EB não otimizada .....</b>	<b>43</b>
<b>Tabela 3.4. Coordenadas dos usuários e da EB para o cenário A. Posição da EB otimizada .....</b>	<b>43</b>
<b>Tabela 3.5. Coordenadas dos usuários e da EB para o cenário B. Posição da EB não otimizada .....</b>	<b>44</b>
<b>Tabela 3.6. Coordenadas dos usuários e da EB para o cenário B. Posição da EB otimizada .....</b>	<b>44</b>
<b>Tabela 4.1. Resultados dos cálculos de perda completa no cenário A. Posição da EB não otimizada .....</b>	<b>50</b>
<b>Tabela 4.2. Resultados dos cálculos de perda completa no cenário A. Posição da EB otimizada .....</b>	<b>50</b>
<b>Tabela 4.3. Resultado dos cálculos de perda completa no cenário B. Posição da EB não otimizada .....</b>	<b>51</b>
<b>Tabela 4.4. Resultado dos cálculos de perda completa no cenário B. Posição da EB otimizada .....</b>	<b>51</b>

## Lista de siglas

- API** Application Programming Interface, Interface de programação de aplicação
- BER** Bit Error Rate, taxa de erro de bit
- BestCQI** Best Channel Quality Indicator, melhor indicador de qualidade do canal
- CQI** Channel Quality Indicator, indicador de qualidade do canal
- CN** Core Network, Núcleo da rede
- CRAN** Cloud Radio Access Network, Rede de acesso de rádio na nuvem
- DL** Downlink, enlace de descida
- EB** Estação-base
- ERP** Effective Radiated Power, Potência efetiva radiada
- FCFS** First Come First Served
- HTTP** Hypertext Transfer Protocol, Protocolo de Transferência de Hipertexto
- IoT** Internet of Things, Internet das coisas
- OFDM** Orthogonal Frequency Divison Multiplexing, multiplexação por divisão de frequências ortogonais
- OSM** Open Street Map
- RAN** Radio Access Network, Rede de acesso de rádio
- RR** Round Robin
- SINR** Signal Interference Noise Ratio, Relação Sinal-interferência mais Ruído
- SIR** Signal Interference Ratio, Relação Sinal-interferência
- SL** System level, nível de sistema
- SNR** Signal Noise Ratio, Relação Sinal-Ruído
- TS** Time slot, intervalo de tempo
- UE** User Equipment, Equipamento do usuário

**UP** Uplink, enlace de subida

## Capítulo 1 – Introdução

Uma rede móvel consiste em um sistema duplex capaz de trocar informações entre uma estação-base (EB) e um dispositivo móvel. Duplex quer dizer que tanto a EB quanto o equipamento móvel podem enviar e receber ao mesmo tempo. Além disso, ao contrário do dispositivo móvel, a EB não se move.

No passado, redes móveis consideravam apenas aparelhos celulares (telefones), e também era chamada de telefonia celular (diferentemente da telefonia móvel), mas com o crescente avanço na internet das coisas (IoT – *Internet Of Things*), diversos outros elementos passaram a utilizar os recursos das redes móveis.

Existem características importantes que diferenciam um sistema de telefonia móvel de um sistema de telefonia celular. Um exemplo de sistema de telefonia móvel é o telefone fixo sem fio, que possui sua base posicionada em um lugar fixo, mas o aparelho telefônico pode se mover dentro da área de cobertura da sua base.

O sistema de telefonia celular tem esse nome devido a sua divisão em células, que são pequenas áreas de cobertura. Os primeiros sistemas telefônicos não eram divididos em clusters e células, e por conta disso não eram tão eficientes. Para grandes cidades, um sistema com apenas 1000 canais apresentava uma baixa disponibilidade do sistema, pois apenas uma fração da população poderia utilizá-lo a cada instante [1].

Após a introdução do conceito de reutilização de frequência, os sistemas se dividiram em células, onde cada célula recebe uma porção dos canais. Um cluster é o nome dado a um conjunto de células. A forma de célula tradicionalmente utilizada em projetos ou em estudos é a célula hexagonal.

Para formação de um cluster, a quantidade de células deve respeitar algumas regras específicas que são apresentadas em detalhes por [1]. Fazendo uso de células e clusters, as frequências disponíveis puderam ser reutilizadas, e essa reutilização das frequências contribuiu para o aumento da capacidade do sistema e disponibilidade do serviço, mas trouxe alguns desafios.

Os principais problemas que surgiram com os sistemas divididos em células foram a interferência e a transferência entre células. Como as frequências passaram a ser reutilizadas, os sinais transmitidos no mesmo canal passaram a interferir um no outro.

Para mitigar os efeitos da interferência, estabeleceu-se uma relação entre o tamanho do cluster a relação sinal-interferência (SIR). Na prática, cada sistema possui seus próprios requisitos, e a relação sinal-interferência mais ruído (SINR – *Signal Interference Noise Ratio*) deve ser grande o suficiente para viabilizar a comunicação.

Em relação a transferência entre células, foi necessário definir uma estratégia para manter a comunicação de um usuário que se encontrava nos limites entre uma célula e outra, denominadas de estratégias de transferência. Essas estratégias levam em conta a intensidade do sinal entre a célula atual e a nova célula candidata e devem determinar o momento mais adequado para transferência sem causar o encerramento de uma chamada, por exemplo.

Ao longo dos anos, as transferências (*handoffs*) tornaram-se mais suaves, e a cada nova geração, espera-se que os sistemas sejam mais eficientes.

## **1.1. Motivação**

Tanto em ambientes rurais quanto urbanos, diversos objetos podem atrapalhar a transmissão. Uma diferença no perfil do terreno entre o usuário e a EB, por exemplo, pode causar perdas na comunicação. Nas cidades, as construções também podem causar perdas na transmissão.

Os resultados apresentados por meio das simulações visam representar e prever o comportamento de redes 5G. Todavia, para que esses resultados estejam mais próximos da realidade, é importante uma análise que considere as especificidades de cada local. Nesse sentido, um simulador popular como o Vienna 5G System Level permite a inclusão de mapas, mas carece de um modelo de perda por difração.

## **1.2. Objetivos**

No presente trabalho, pretende-se avaliar o efeito da perda por difração em dois cenários: No Campus Santa Mônica da Universidade Federal de Uberlândia e no Campus Glória, da mesma universidade, através de um simulador que considera apenas a perda básica.

Além disso, objetiva-se analisar os impactos da alteração da posição da estação-base nos valores de perda e no desempenho da rede.

## **1.3. Organização do trabalho**

O trabalho está organizado da seguinte forma. No capítulo 2 é apresentada a base teórica, com a apresentação de alguns conceitos utilizados para o desenvolvimento das equações e dos códigos utilizados neste trabalho. No capítulo 3 são apresentadas as metodologias utilizadas no desenvolvimento do software para o cálculo de perda por difração e introduzidos os cenários de simulação. Em seguida, no capítulo 4 são apresentados os resultados dos cálculos e das simulações, e finalmente, no capítulo 5 mostram-se as conclusões a respeito dos resultados obtidos e sugestões de melhorias que podem ser implementadas.

## **Capítulo 2 – Fundamentação Teórica**

### **2.1. Propagação de rádio móvel**

#### **2.1.1. Introdução à propagação de ondas**

A propagação de ondas eletromagnéticas em meios não guiados pode trazer maiores dificuldades em comparação com a propagação em guias de onda. Isso pode ocorrer por conta da maior variabilidade das condições do ambiente de propagação.

Existem diversos mecanismos básicos de propagação. Os principais são a reflexão, difração e dispersão [1]. Em qualquer cenário de propagação terrestre, pelo menos um desses mecanismos estará presente.

Durante a transmissão, uma parte da energia do sinal é perdida por conta da interação com o meio. Nos cenários de transmissão sem fio, essas perdas tendem a ser mais significativas, visto que a energia se distribui em uma região maior. Além disso, existem diversas fontes de ruído e interferência que levam a degradação do sinal [1].

Para que um sistema de transmissão sem fio seja viável, existem alguns fatores importantes. Desconsiderando a difração, a reflexão e a dispersão, e considerando inicialmente apenas a perda básica, percebe-se a necessidade de direcionar o máximo da energia da onda eletromagnética ao receptor. Isso é feito utilizando antenas diretivas, que favorecem a transmissão em uma determinada direção.

A escolha das antenas afeta diretamente a qualidade do sinal recebido, visto que antenas mais diretivas somente beneficiarão receptores que se encontram alinhados. Da mesma forma, antenas pouco diretivas distribuirão a energia em uma região maior, reduzindo a qualidade da transmissão, sobretudo em aspectos tratados na seção 2.2.

O fenômeno que ocorre quando uma onda eletromagnética incide sobre um objeto que possui dimensões grandes quando comparado com o comprimento da onda em propagação, é denominado de reflexão. No meio urbano, essas reflexões acontecem em paredes, prédios, terraços e outras estruturas semelhantes.

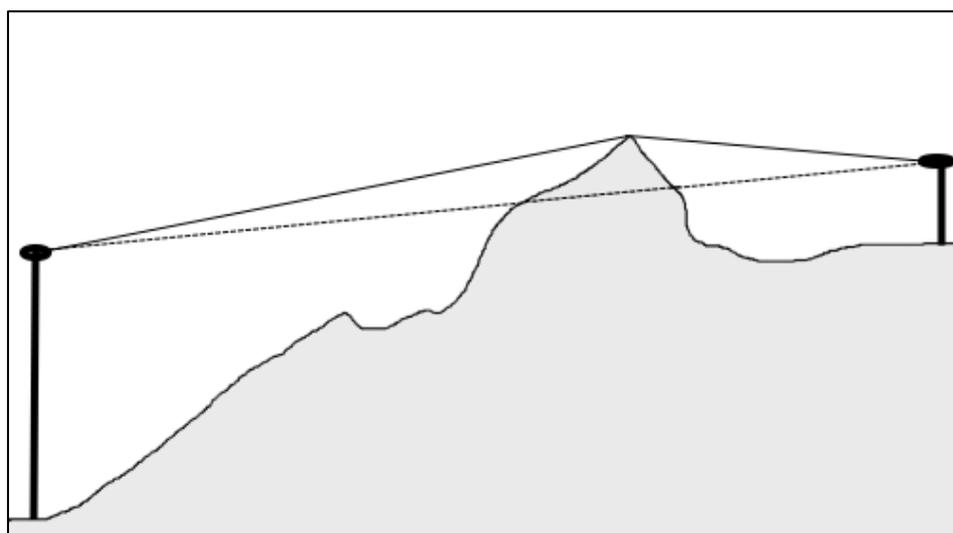
O resultado da reflexão é a mudança da direção de propagação de uma onda eletromagnética. Isso geralmente leva a uma redução significativa da amplitude do campo eletromagnético refletido, dependendo do coeficiente de reflexão.

Em alguns casos, a reflexão pode fazer a onda eletromagnética refletida chegar ao receptor, fenômeno conhecido como *múltiplo percurso*. Como efeito, pode ocorrer uma interferência construtiva, onde a onda eletromagnética refletida e a onda eletromagnética direta chegam em fase no receptor, e, com isso, ocorre um aumento na amplitude do campo elétrico recebido.

Por outro lado, o cenário mais provável é que ocorra uma interferência destrutiva, onde as ondas eletromagnéticas refletida e direta não chegarão em fase ao receptor, levando à diminuição da amplitude do campo elétrico recebido, e, por consequência, reduzindo a qualidade da transmissão devido a degradação da relação sinal-ruído (SNR – *Signal Noise Ratio*).

A difração é um dos problemas na propagação de ondas em sistemas de telefonia móvel celular. Baseada no princípio de Huygens, que estabelece que cada frente de onda atua como uma nova fonte de onda, gerando novas ondas [2], ocorre quando o caminho entre o transmissor e o receptor se encontra obstruído por alguma irregularidade na superfície [1], conforme apresentado na figura 2.1.

Figura 2.1. Representação da difração



Fonte: o Autor.

As obstruções que fazem as ondas eletromagnéticas sofrerem difração são, muitas vezes, irregularidades consequentes das características não uniformes do perfil do terreno, considerando a posição dos receptores móveis em relação ao transmissor.

A depender das zonas de Fresnel obstruídas, pode haver uma diminuição significativa da potência recebida, prejudicando a comunicação por conta da deterioração da SNR. Em alguns casos, ainda pode ser possível a troca de informações, mas em uma taxa consideravelmente menor.

Quando a superfície apresenta certas irregularidades, ou o meio em que a onda trafega é composto por objetos cujas dimensões sejam consideradas pequenas quando comparadas ao comprimento de onda, e a quantidade de obstáculos seja comparavelmente grande, ocorre o fenômeno chamado dispersão.

Objetos como folhagens, postes de iluminação e outros elementos podem induzir a dispersão das ondas eletromagnéticas.

### **2.1.2. Modelos e perdas básicas de propagação**

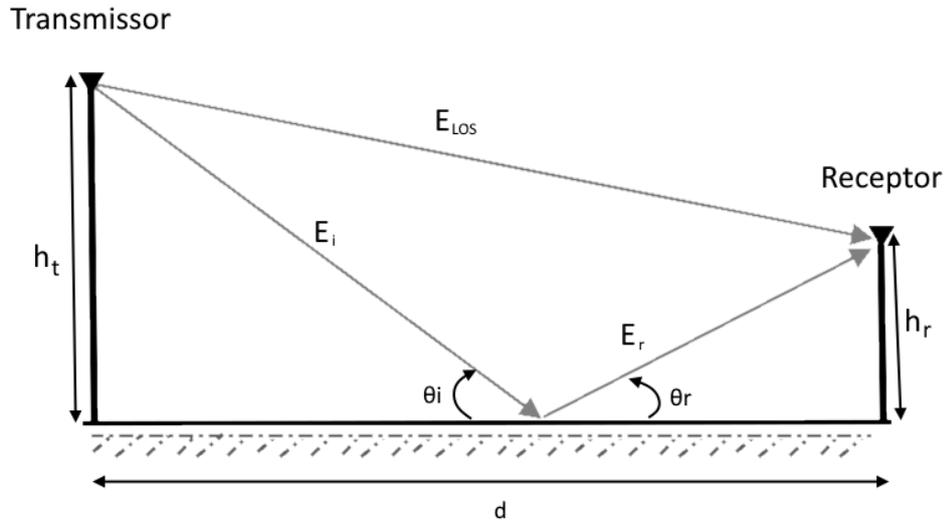
Neste trabalho, são abordados dois modelos de perda básica: o modelo de dois raios e o modelo de perda básica do espaço livre, utilizando a equação de Friis. Esses modelos foram escolhidos baseando-se no modelo de Durkin, que é um modelo desenvolvido para sistemas celulares [1].

#### **Modelo de dois raios**

Também conhecido como modelo de reflexão no solo, o modelo de dois raios considera a potência recebida como sendo a resultante de uma onda direta somada à uma onda refletida no solo. Esse modelo é justificado pelo fato de, na prática, ser muito raro uma ocasião onde um caminho direto é o único caminho de propagação.

Os resultados do modelo são obtidos considerando a ótica geométrica. A partir da geometria mostrada na figura 2.2 e com a aplicação da teoria das imagens, tem-se algumas equações úteis resultantes da geometria [1].

Figura 2.2. Geometria para o modelo de dois raios.



Fonte: o Autor.

Os símbolos apresentados na figura 2.2 possuem os seguintes significados:

- $E_{LOS}$  representa o campo elétrico que segue em uma trajetória direta do transmissor ao receptor, ao longo da linha de visada (LoS).
- $E_i$  e  $E_r$  representam o campo elétrico incidente e refletido, respectivamente.
- $\theta_i$  e  $\theta_r$  representam os ângulos de incidência e reflexão, respectivamente.
- $h_t$ ,  $h_r$  e  $d$  representam a altura do transmissor, a altura do receptor e a distância entre o transmissor e o receptor, respectivamente.

Duas equações utilizadas no desenvolvimento do modelo são a diferença do caminho  $\Delta$  e a diferença de fase  $\theta_\Delta$ .

$$\Delta = \sqrt{(h_t + h_r)^2 + d^2} - \sqrt{(h_t - h_r)^2 + d^2}, \quad (2.1)$$

e

$$\theta_{\Delta} = \frac{2\pi\Delta}{\lambda}. \quad (2.2)$$

A equação 2.3 é utilizada para calcular a potência recebida baseada no modelo de dois raios.

$$P_r(d) = P_t G_t G_r \frac{h_t^2 h_r^2}{d^4}. \quad (2.3)$$

Onde:

- $P_r$  representa a potência recebida.
- $P_t$  representa a potência transmitida.
- $G_t$  representa o ganho da antena transmissora.
- $G_r$  representa o ganho da antena receptora.
- $h_t$  e  $h_r$  representam as alturas da antena transmissora e receptora, respectivamente.
- $d$  representa a distância de separação do transmissor e do receptor.

### **Equação de Friis**

A equação de Friis é uma expressão mais geral para o cálculo de perda. Ela relaciona a potência recebida por uma antena com a transmitida em outra, separadas por uma distância maior que duas vezes o quadrado da maior dimensão de qualquer uma das antenas, dividido pelo comprimento de onda [3]. Sua utilização é mais comum no projeto de enlaces de satélites.

$$P_r = \frac{P_t G_t G_r \lambda^2}{(4\pi R)^2 L}. \quad (2.4)$$

As variáveis apresentadas na equação 2.4 possuem os seguintes significados:

- $P_t$  e  $P_r$  representam a potência transmitida e recebida, respectivamente.
- $G_t$  e  $G_r$  representam os ganhos das antenas transmissora e receptora, respectivamente.
- $\lambda$  representa o comprimento de onda.
- $R$  representa a distância de separação entre as antenas.
- $L$  representa o fator de perdas do sistema, e não está relacionado com a propagação.

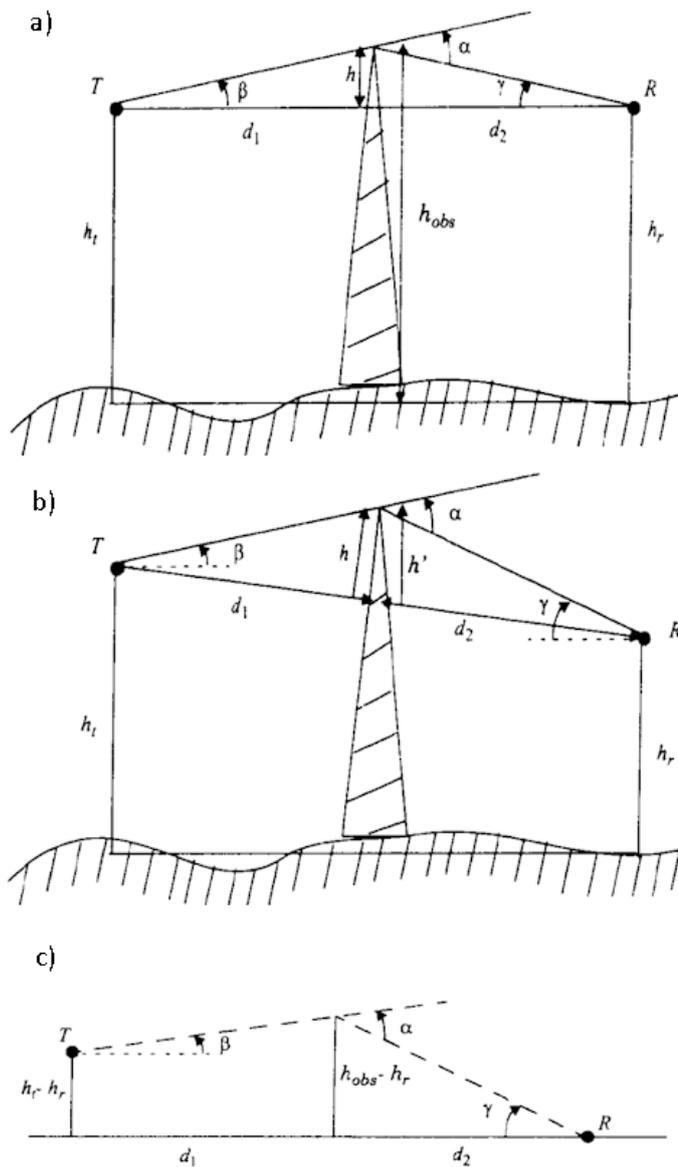
Em alguns casos, considera-se  $L = 1$ . A expressão final para a potência recebida utilizando a equação de Friis, considerando  $L = 1$  pode ser vista na equação 2.5.

$$P_r = \frac{P_t G_t G_r \lambda^2}{(4\pi R)^2}. \quad (2.5)$$

### 2.1.3. Perda por difração

Caso haja, é importante considerar a perda por difração no projeto de um enlace. Se essa perda for muito grande, pode atrapalhar o correto funcionamento do sistema. A figura 2.3 mostra a difração em diferentes geometrias, auxiliando na compreensão das equações. A partir da geometria da figura 2.3b, a diferença entre o caminho direto e o caminho difratado pode ser obtida, conforme mostra a equação 2.6.

Figura 2.3. Geometria da difração do gume de faca. a) Transmissor e receptor na mesma altura. b) Transmissor e receptor em alturas diferentes. c) Simplificação de b) com uma geometria equivalente, subtraindo a altura menor ( $h_r$ ).



Fonte: [1].

$$\Delta = \frac{h^2 (d_1 + d_2)}{2 d_1 d_2} . \quad (2.6)$$

A diferença de fase pode ser calculada a partir da equação 2.7.

$$\phi = \frac{2\pi\Delta}{\lambda} = \frac{2\pi h^2 (d_1 + d_2)}{\lambda 2 d_1 d_2}. \quad (2.7)$$

Sabendo que para um valor  $x$  pequeno,  $t g x \approx x$ . Sendo  $\alpha = \beta + \gamma$  (geometria da figura 2.3c), a equação 2.8 é obtida:

$$\alpha = h \left( \frac{d_1 + d_2}{d_1 d_2} \right). \quad (2.8)$$

Normalizando a equação 2.7 usando o parâmetro de difração sem dimensão de Fresnel-Kirchoff, dado pela equação 2.9, a equação 2.10 é obtida.

$$v = h \sqrt{\frac{2(d_1 + d_2)}{\lambda d_1 d_2}} = \alpha \sqrt{\frac{2d_1 d_2}{\lambda(d_1 + d_2)}}. \quad (2.9)$$

A equação 2.10 expressa a diferença de fase em função do parâmetro  $v$ .

$$\phi = \frac{\pi}{2} v^2. \quad (2.10)$$

As zonas de Fresnel contribuem de maneira alternada para interferência destrutiva e construtiva. Um parâmetro que é de grande importância na implementação deste trabalho é o raio da zona de Fresnel, denotado por  $r_n$ , onde  $n$  especifica qual a zona de Fresnel (primeira, segunda, etc).

$$r_n = \sqrt{\frac{n\lambda d_1 d_2}{d_1 + d_2}}. \quad (2.11)$$

A primeira zona de Fresnel concentra a maior parte da energia da onda eletromagnética. Portanto, neste trabalho foi considerada apenas a obstrução da primeira zona de Fresnel.

As equações 2.12a, 2.12b, 2.12c, 2.12d e 2.12e foram utilizadas para o cálculo de ganho por difração.

$$G_d(dB) = 0, v \leq -1; \quad (2.12a)$$

$$G_d(dB) = 20 \log(0,5 - 0,62v), -1 \leq v \leq 0; \quad (2.12b)$$

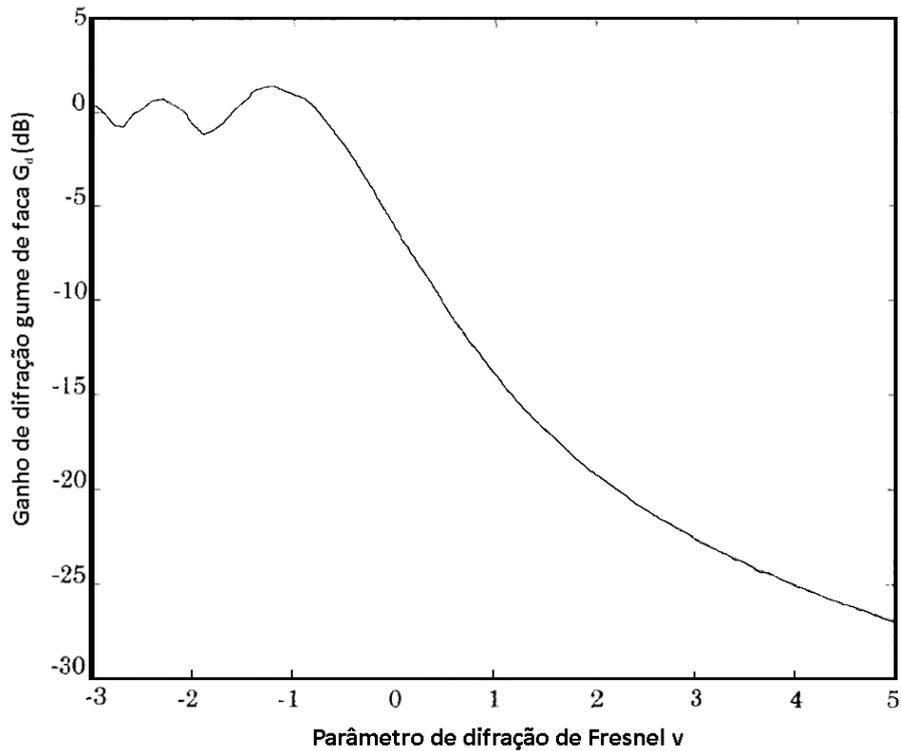
$$G_d(dB) = 20 \log(0,5 \exp(-0,95v)), 0 \leq v \leq 1; \quad (2.12c)$$

$$G_d(dB) = 20 \log\left(0,4 - \sqrt{0,1184 - (0,38 - 0,1v)^2}\right), 1 \leq v \leq 2,4; \quad (2.12d)$$

$$G_d(dB) = 20 \log\left(\frac{0,225}{v}\right), v > 2,4. \quad (2.12e)$$

Em alguns casos, um gráfico também é utilizado na determinação do ganho por difração. Esse gráfico pode ser visto na figura 2.4.

Figura 2.4. Ganho da difração do tipo gume de faca em função do parâmetro  $v$ .



Fonte: Adaptado de [1].

## 2.2. Comunicações digitais

### 2.2.1. Introdução

Os receptores digitais estão presentes em praticamente todos os transceptores modernos. O foco desta seção é apresentar os conceitos gerais de um receptor digital, como os utilizados em aparelhos móveis celulares, sua sensibilidade ao ruído e a importância de minimizar as perdas nos receptores digitais.

### 2.2.2. Arquitetura típica de um receptor digital

Existem vários tipos de receptores digitais. De forma geral, a maioria deles inicia o processo de recepção por meio de uma antena, onde o sinal chega ao receptor. Após isso, o sinal recebido pode passar por um filtro passa-banda, ou por um *mixer*, caso a frequência de trabalho seja diferente da frequência de transmissão, como ocorre nas mudanças de canais.

Essa é a descrição básica do ponto de entrada de um receptor, ainda em termos de radiofrequência, que entra em aspectos de como o sinal é modulado, como é transmitido, qual a largura de banda do filtro, etc.

Mesmo que eles possuam os mesmos princípios, as arquiteturas dos receptores digitais geralmente diferem entre si, visto que em alguns, a modulação é controlada e pode ser alterada, enquanto outros trabalham em apenas um tipo de modulação.

O objetivo dessa seção é apresentar o receptor digital do ponto de vista da detecção, ou seja, no momento em que o detector determina qual foi o dígito recebido.

No caso de um receptor binário, quando o sinal se encontra no ponto de amostragem, o detector determina a qual intervalo esse valor amostrado pertence e com isso determina o dígito correspondente.

Supondo que o valor amostrado é maior ou igual a zero, o receptor informa em sua saída que o dígito recebido foi **1**. Todavia, caso o valor amostrado seja inferior a zero, o receptor diz que o dígito recebido foi **0** [4].

Esse é o propósito de um receptor digital. Entretanto, existe um grande problema: o ruído. Com o ruído aditivo (que se soma ao sinal), o valor original pode até ser inferior ao limiar de amostragem, mas a tensão de ruído somada ao sinal original leva a saída amostrada a um valor superior ao limiar de amostragem, configurando um erro de detecção.

Por conta desses erros de detecção, são utilizados códigos corretores de erros, que atuam auxiliando na correção dos bits que foram detectados de forma incorreta. A estratégia utilizada em boa parte dos receptores para reduzir a quantidade de erros de detecção é a chamada adaptação de taxa.

Essa estratégia torna os transceptores capazes de alterarem sua modulação para uma modulação mais resistente ao ruído quando a taxa de erro de bit (BER – *Bit Error Rate*) se encontra elevada, onde há uma quantidade menor de limiares de decisão (valores de referência para comparação com o sinal amostrado).

Resumidamente, apenas alterando o tipo de modulação, o transmissor e o receptor podem reduzir significativamente a BER, tornando possível a comunicação.

### **2.2.3. Impactos da relação sinal-ruído na recepção de um sinal**

Como explicado em seções anteriores, quando o receptor se afasta do transmissor, a potência recebida também diminui. Isso leva à uma redução da SNR e, conseqüentemente, aumenta a BER da detecção, levando o receptor e o transmissor a alterarem a modulação utilizada com o intuito de reduzir a BER.

É importante considerar que quanto mais símbolos uma determinada modulação possui, maiores serão as taxas de transmissão. Por consequência, quando há necessidade de alterar a modulação de uma com maior quantidade de símbolos para uma com menor quantidade de símbolos, a taxa de transmissão será reduzida.

Isso é perceptível nos telefones, computadores e outros dispositivos que possuem receptores que realizam essa função. À medida que o usuário se afasta do transmissor, geralmente ocorre uma redução significativa da vazão. Isso pode ser visto no WiFi, por exemplo.

Considerando um cenário no qual um usuário de um serviço de telefonia móvel percorre um determinado caminho, e neste caminho há uma variação considerável no perfil do terreno de tal forma que ocorra difração, isso causa um impacto na qualidade do sinal.

A difração leva a uma queda na potência recebida, mesmo que os pontos em que o usuário tenha se posicionado sejam equidistantes à EB [1].

Em outras palavras, a difração também diminui a SNR, que ocasiona os mesmos problemas descritos anteriormente, levando o receptor e o transmissor a mudarem suas

modulações para alguma que tenha uma quantidade menor de símbolos, reduzindo a vazão do usuário.

### 2.3. Visão geral do 5G

A tecnologia do 5G já vem sendo utilizada há algum tempo em alguns países, mas no Brasil essa tecnologia ainda não está disponível em todas as regiões. Entretanto, diversas operadoras tem trabalhado para aumentar a disponibilidade desta tecnologia em todo o país.

O objetivo do 5G é entregar maiores taxas, menores latências e prover conectividade em massa, garantindo uma estrutura de rede com maior qualidade de serviço. Isso traz uma grande contribuição para a IoT, facilitando o uso de sensores e a automatização e o controle de sistemas.

O 5G faz uso da multiplexação por divisão de frequências ortogonais (OFDM – *Orthogonal Frequency Division Multiplexing*). O conceito de ortogonalidade está relacionado com a não sobreposição de frequências, reduzindo as fontes de interferência [4].

As taxas máximas esperadas estão em torno de 2.5Gbps para *download* e 1.25Gbps para *upload* [7]. Além disso, a arquitetura do 5G foi planejada para utilizar uma rede de acesso de rádio na nuvem (CRAN – *Cloud Radio Access Network*).

Em termos de arquitetura, existem três componentes principais, sendo eles:

- Equipamento do usuário (UE – *User Equipment*), que é o dispositivo que se conecta à rede. Pode ser um equipamento de telefone ou alguma espécie de sensor.
- Rede de acesso a rádio (RAN – *Radio Access Network*), é responsável por prover a conectividade entre o usuário e o núcleo da rede. Geralmente, inclui a EB, as antenas e quaisquer outros equipamentos relacionados que possam prover cobertura de rádio na região em questão.
- Núcleo da rede (CN – *Core Network*), responsável por gerenciar e direcionar o tráfego entre o usuário e a internet.

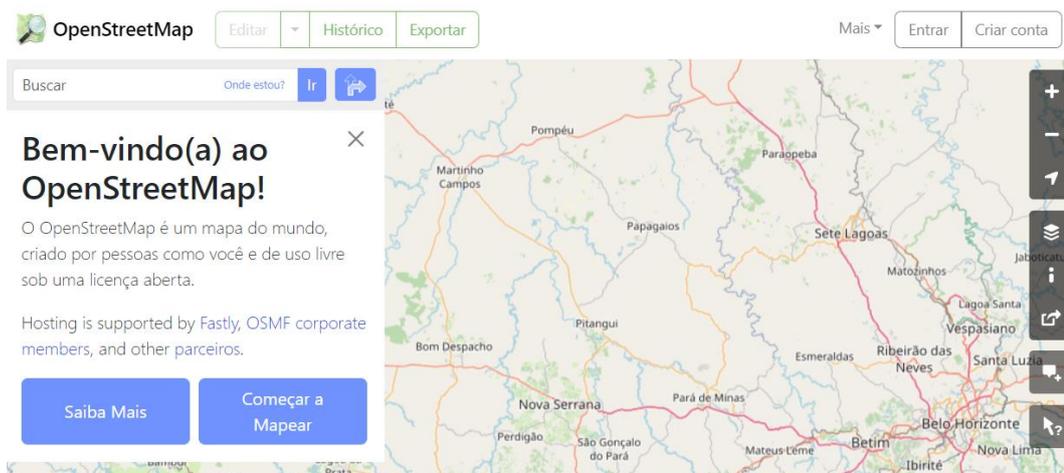
Outra técnica interessante do 5G é a utilização do *beamforming*, que consiste em controlar o feixe da antena, direcionando o sinal para o equipamento do usuário, aumentando a força do sinal (maior diretividade por parte da antena) e reduzindo a interferência. Isso melhora a taxa de transferência e a latência (consequência da melhor SNR, por conta do aumento da potência).

## 2.4. Simulador Vienna 5G System Level

### 2.4.1. Introdução

O simulador Vienna 5G SL (desenvolvido por TU Wien) é um simulador de redes com muitas funcionalidades. Em termos gerais, é um simulador capaz de simular redes com grandes quantidades de estações-base e vários usuários. Ele é capaz de realizar simulação de diversos cenários e é possível utilizar o OpenStreetMap (OSM) para projetar cenários em torno de locais específicos, além de considerar a perda que as paredes ou edificações presentes nestes locais podem causar.

Figura 2.5. Página do OpenStreetMap



Fonte: [13]

Para utilizar o OSM neste simulador, uma forma simples consiste em alterar a região, ajustando os parâmetros apresentados na figura 2.6.

Figura 2.6. Trecho do código responsável por definir a região, no cenário *openStreetMap* do simulador.

```
% The following parameters describe the specification of an OpenStreetMap
% city. The desired real-world area is specified via latitudes and
% longitudes, which can easily be obtained from openstreetmap.com via the
% "Export" feature.
openStreetMapCity = parameters.city.OpenStreetMap;
openStreetMapCity.latitude      = [48.1955904, 48.1973271];
openStreetMapCity.longitude    = [16.3690209, 16.3718059];
openStreetMapCity.streetWidth  = 5;
% All buildings have random heights, which are sampled from a uniform
% distribution bounded by the minimum and maximum building height. For
% reproducibility it is possible to specify an integer seed for the height
% random number generator.
openStreetMapCity.minBuildingHeight = 10;
openStreetMapCity.maxBuildingHeight = 25;
openStreetMapCity.heightRandomSeed = 'shuffle';
openStreetMapCity.wallLossdB       = 10;
```

Fonte: captura de código realizada pelo Autor.

Algumas medidas de desempenho importantes como, por exemplo, vazão, SINR, latência, etc, podem ser obtidas a partir do simulador, que também conta com modelos de propagação no interior de ambientes (*indoor*), realizando uma estimativa das perdas geradas por construções. Além disso, estão disponíveis modelos para tráfego veicular, de pedestres, usuários em ambientes internos, modelos de antenas e muito mais.

#### **2.4.2. Simulação no simulador Vienna 5G SL**

O Simulador Vienna 5G SL utiliza o MATLAB® como parte de sua estrutura. Para utilizá-lo, são necessárias algumas ferramentas do MATLAB® (toolbox). Para realizar uma simulação, dois elementos principais são necessários: um lançador e um cenário. Estes elementos pertencem a estrutura do simulador.

Tanto o lançador quanto o cenário podem ser modificações dos exemplos disponibilizados. Geralmente, no lançador são definidos os resultados que devem ser coletados, e no cenário toda a configuração do ambiente é desenvolvida, como a quantidade de usuários, as posições, a quantidade de estações-base, a densidade, o tipo de antena, o modelo de canal, modelo de perda, etc.

A estrutura do simulador divide-se em quatro partes principais: inicialização, pré-geração, *loop* de simulação principal e pós-processamento. Alguns elementos importantes do simulador são: *time slots* (TS), segmentos (*segments*) e *chunks*.

Na inicialização, todos os parâmetros necessários são definidos antes de começar a simulação, principiando a *timeline* do simulador, que é o que divide os elementos citados acima.

O TS é uma unidade de tempo de valor fixo baseado em qual agendamento está sendo executado. Um segmento (*segment*) é definido como o tempo durante o qual os valores do desvanecimento macroscópico são aproximadamente constantes; as *chunks* são conjuntos de segmentos.

Na etapa de pré-geração, todos os elementos que pertencem a algum local geométrico são criados, como cenários tridimensionais ou representações físicas de objetos. Existe a opção de descartar esses objetos tridimensionais e realizar simulações bidimensionais [5].

Nesta mesma etapa de pré-geração, após a criação dos elementos físicos, as estações-base e os usuários são gerados de acordo com os parâmetros definidos anteriormente. Concluída essa etapa, as informações necessárias são salvas, e as informações importantes para cada *chunk* são combinadas. A maior vantagem disso é a redução de dados não essenciais.

No *loop* principal de simulação, cada *chunk* é processada, e com isso, cada TS também é processado. Nas próximas etapas, o simulador realiza verificações no TS para averiguar se o mesmo marca o início de um novo segmento.

Para o início de um novo segmento, as posições de todos os usuários são atualizadas, assim como os valores de desvanecimento macroscópico para todos os elementos transmissores e todos os interferentes [5].

Por fim, quando todas as *chunks* são calculadas, os resultados individuais são combinados e tem-se então os resultados finais, que podem ser salvos à critério do utilizador.

### 2.4.3. Modelos utilizados do simulador

O modelo de antena utilizado para a EB foi a de três setores. Diversos parâmetros podem ser configurados. Alguns deles são: azimute e elevação, que são os ângulos nos quais a antena tem o ganho máximo; o tipo de estação base; altura; quantidade de receptores; quantidade de transmissores, etc.

Para o modelo de canal, foi escolhido o modelo *Pedestrian A*. De acordo com [6], tem-se as informações do perfil de intensidade de multi-percurso, que apresenta alguns valores de atraso em função da potência. Essas informações podem ser vistas na tabela 2.1.

Tabela 2.1. Modelo de canal ITU *Pedestrian A*.

Atraso (ns)	0	110	190	410
Potência (dB)	0.0	9.7	19.2	22.8

Fonte: Adaptado de [6].

## 2.5. Considerações Finais

Neste capítulo, foram apresentados os fundamentos teóricos para o desenvolvimento do trabalho e uma visão geral de algumas ferramentas utilizadas. Alguns dos tópicos deste capítulo contribuirão para a compreensão e interpretação dos resultados do simulador.

Com a compreensão dos fenômenos físicos que acontecem com as ondas eletromagnéticas, como os conceitos de propagação e difração, objetiva-se analisar o impacto que perdas resultantes da propagação podem causar em etapas posteriores, como na etapa de demodulação do sinal.

Algo interessante apresentado na seção 2.4, que traz uma visão geral das redes 5G, é a utilização de antenas inteligentes. Isso remete ao tópico 2.1, que mostra a importância da diretividade da antena na qualidade do sinal (equações 2.3 e 2.4).

Neste sentido, espera-se que as antenas inteligentes sejam capazes de reduzir os efeitos indesejáveis consequentes da propagação, maximizando a potência recebida, e reduzindo a interferência por conta da sua alta diretividade apenas na direção desejada, sem afetar outras direções.

## Capítulo 3 – Metodologia

### 3.1. Introdução

Para a realização deste trabalho, um software foi desenvolvido em *python* [Apêndices A e B], para realizar os cálculos de perda de caminho e por difração. O software considerou o pior valor possível da perda por difração, respeitando as limitações do modelo utilizado. Além disso, foi desenvolvido um algoritmo para otimização da posição da EB.

O software original para o cálculo de perda foi desenvolvido em *javascript*, para um trabalho da disciplina de comunicações móveis na Universidade Federal de Uberlândia. A versão do software em *python* foi desenvolvida para fins de praticidade.

Os dados do perfil do terreno podem ser obtidos de duas formas. A primeira é a partir de uma interface de programação de aplicação (API – *Application Programming Interface*) gratuita, chamada *Open-Elevation* [8]. A outra API que pode ser utilizada é a *Google Elevation* [9].

Tanto a *Open-Elevation* quanto a *Google Elevation* são boas opções, mas a API do Google fornece os dados com uma maior precisão para curtas distâncias, além de ser capaz de interpolar os pontos intermediários entre o transmissor e o receptor, entregando as alturas de acordo com a quantidade de amostras solicitadas.

Essas duas APIs trabalham com requisições do protocolo de transferência de hipertexto (HTTP – *Hypertext Transfer Protocol*). Elas suportam as requisições do tipo GET HTTP, e a *Open-Elevation* é capaz de suportar do tipo POST [10].

A API do Google possui algumas vantagens em relação a *Open-Elevation*. Por exemplo, se forem solicitadas 100 amostras na requisição a API do Google, ele retornará um vetor com informações dos 100 pontos, como latitude, longitude e elevação.

Entretanto, não é possível solicitar as mesmas 100 amostras na requisição da API *Open-Elevation*, ao menos, não de forma direta. Na API do Google, os dados enviados na requisição são: latitude e longitude do ponto inicial, latitude e longitude do ponto final, quantidade de amostras e chave de API.

Na *Open-Elevation*, por se tratar de uma API gratuita, não é necessário enviar nenhuma chave, mas todos os pontos devem ser enviados, latitude e longitude do ponto 1, latitude e longitude do ponto 2, até o ponto  $n$ . Por conta disso, foi necessário implementar um algoritmo para interpolar os pontos.

Como as distâncias em células de sistemas de telefonia móvel são curtas quando comparadas as distâncias em enlaces de microondas, especialmente nos casos avaliados neste trabalho, uma interpolação linear foi suficiente.

A API gratuita (*Open-Elevation*) apresenta resultados melhores e mais próximos aos da API do Google quando as distâncias são na ordem de dezenas de quilômetros. Para curtas distâncias, os resultados obtidos com a API gratuita também são aceitáveis, mas os dados de elevação são menos precisos.

No desenvolvimento do código, um ambiente virtual foi configurado para a inclusão das bibliotecas e arquivos necessários. Esse ambiente virtual faz parte da biblioteca *virtualenv* do *python*, e pode ser instalado com o comando *pip install virtualenv*.

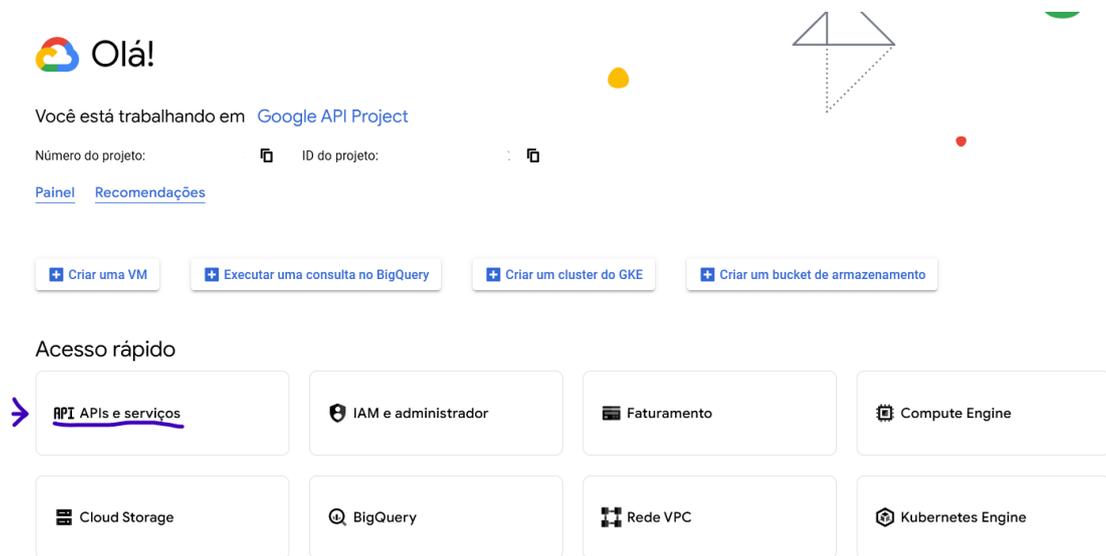
A vantagem de utilizar um ambiente virtual é que as bibliotecas instaladas no ambiente virtual não interferem no conjunto de bibliotecas da instalação principal do *python*. Isso é bom porque fornece mais controle ao usuário sobre as bibliotecas utilizadas, além de ser um ambiente mais seguro e fácil de trabalhar.

Outra biblioteca importante utilizada foi a **lib** *decouple*, para obter valores de variáveis armazenadas em um arquivo *.env*. A utilização deste tipo de biblioteca e arquivos *.env* é comum, sobretudo quando algum software faz a utilização de diversas APIs que precisam de uma chave para autenticação.

Ao invés de salvar chaves em variáveis dentro do código, especialmente nos casos em que essas variáveis dão acesso a APIs pagas ou com informações sensíveis, a melhor alternativa é armazenar essas chaves em alguma variável de ambiente, ou em um ambiente virtual, como foi realizado neste trabalho.

No arquivo *.env*, uma variável chamada “GOOGLE\_KEY” foi salva, e o valor atribuído foi a chave de API obtida no *Google Cloud Platform* [11]. Para acessar o menu de chaves, basta navegar até “APIs e serviços”, conforme pode ser visto na figura 3.1.

Figura 3.1. Página principal do *Google Cloud Platform*.



Fonte: Captura de [11] realizada pelo Autor.

Após isso, o usuário terá acesso as principais bibliotecas e APIs do Google, e será capaz de gerar chaves de API para utilizar em projetos com este. É importante observar que isso possui um custo, sendo necessário configurar uma forma de pagamento.

O Google disponibiliza um período de testes gratuito e uma certa quantia de créditos para testes, mas isso é disponibilizado apenas por um pequeno período, somente para a realização dos testes e configuração do software.

Após o fim do período gratuito, caso o utilizador deseje prosseguir com o uso das APIs, é necessário configurar um plano de uso e faturamento. Por conta disso, a opção da *Open-Elevation* foi adicionada, para dar ao usuário a chance de utilizar o software sem precisar pagar ou configurar um plano de faturamento.

### 3.2. Software para cálculo de perda

Em relação ao algoritmo para cálculo de perda por difração, existem diversos modelos desenvolvidos por vários autores, como Bullington, Deygout, Epstein-Peterson e outros.

Neste software, foi feita uma adaptação do modelo de Durkin, que é um modelo próprio para sistemas celulares [1].

Outra observação a respeito do modelo utilizado é que a curvatura da Terra não foi levada em consideração, por conta da pequena distância entre o transmissor e o receptor, que torna desprezível a diferença por seu uso [12].

Para o cálculo da perda por difração, o software realiza uma varredura procurando por obstáculos ao longo do perfil do terreno. Para isso, o raio da primeira zona de Fresnel é calculado em cada ponto da linha de visada.

Para obter a componente vertical correspondente, algumas correções na equação 2.11 foram realizadas. Isso se deve ao fato da linha de visada nem sempre ser paralela à superfície (considerando uma superfície plana).

É possível que em alguns casos a linha de visada não tenha nenhuma inclinação (quando a soma das alturas do perfil do terreno + altura do transmissor é igual à soma das alturas do perfil do terreno + altura do receptor em seus respectivos pontos). No entanto, não é isso que acontece na maioria dos casos.

Estabelecida a linha de visada entre o transmissor e o receptor e calculado o raio da primeira zona de Fresnel em cada ponto, uma varredura é realizada em cada ponto, verificando se algum ponto do perfil do terreno invade alguma parte da primeira zona de Fresnel.

Caso haja uma obstrução apenas, aquele ponto é tratado como um gume de faca, a perda consequente à essa obstrução é calculada e finalmente, o resultado final é apresentado. Entretanto, caso haja mais de um ponto, o algoritmo se torna mais complexo.

Avaliando a situação com mais pontos de obstrução, o algoritmo realiza uma varredura procurando pelo ponto mais significativo, utilizando sua altura ou coeficiente angular como critério. Com esse ponto, duas novas retas são traçadas: uma que liga o transmissor a esse ponto, e outra que liga o receptor a esse ponto.

Após isso, uma nova varredura é realizada, verificando se algum obstáculo obstrui algum ponto dessas novas linhas. Caso não haja nenhum obstáculo significativo, calcula-se a perda pelo maior ponto.

No entanto, se houver algum outro obstáculo significativo, calcula-se o obstáculo equivalente entre eles e o algoritmo é aplicado novamente, até que os obstáculos se reduzam em apenas um. Quando isso acontece, a perda é calculada considerando esse obstáculo equivalente, e apresenta-se o resultado final.

Observando o trabalho de pesquisadores que atuam nessa área e suas formas de calcular a perda por difração, é esperado que os resultados obtidos sejam mais otimistas, ou seja, a perda real certamente será maior que a estimada.

### **3.3. Software para otimizar a posição da estação-base**

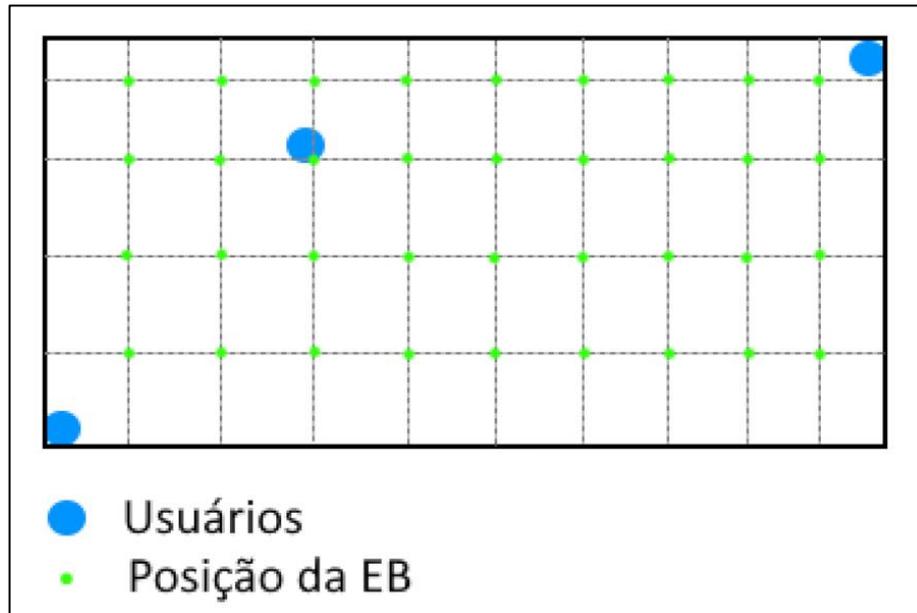
Considerando o cenário mais simples possível, com usuários estáticos, uma maneira de escolher a posição da EB é encontrar o local onde a perda entre ela e cada usuário seja a menor possível.

A forma de otimização realizada neste trabalho foi feita considerando que os usuários estejam estáticos, e a metodologia empregada dividiu o problema da seguinte forma:

Dada uma matriz de usuários, cada um com sua posição (latitude e longitude), foi criada uma área ao redor deles, onde todos os usuários se encontram dentro dessa área. Após isso, a área foi dividida em vários pontos, de maneira proporcional à sua largura e altura.

Isso quer dizer que se a largura dessa região em que os usuários se encontram for maior que a altura dessa mesma região, então mais pontos serão distribuídos no eixo horizontal. Porém, o caso contrário também pode ocorrer.

Figura 3.2. Ilustração do software de otimização da posição da EB.



Fonte: o Autor.

Em outras palavras, se a altura dessa área for maior que a largura, então mais pontos serão distribuídos ao longo do eixo vertical. Caso as dimensões sejam aproximadamente iguais para a altura e a largura, então os pontos serão distribuídos igualmente, criando uma matriz quadrada.

Após a criação da matriz de pontos que armazenarão as posições (latitude e longitude) da EB em cada ponto da matriz, o próximo passo será realizar o cálculo de perda completa.

Devido a grande quantidade de pontos obtidos, essa etapa de otimização pode ser bem lenta. Além disso, é preciso ter cuidado para não atingir o limite das APIs, pois caso isso aconteça, o programa não será capaz de prosseguir para obter os resultados.

Com o cálculo da perda entre cada usuário e a EB em cada posição da matriz, um novo vetor é calculado, armazenando o valor médio das perdas. É uma média simples entre as perdas da EB e cada usuário, para cada posição.

Com o vetor de perdas calculado, a posição otimizada é determinada com base na menor perda média do vetor, ou seja, esse algoritmo otimiza a posição da EB de acordo com a menor perda média. Dessa forma, pode ser que para uma quantidade de usuários  $n = m + o$ ,  $m$  usuários tenham uma perda muito menor que os demais  $o$  usuários.

Existem diversas formas de realizar a otimização da posição da EB, que serão abordadas na seção de trabalhos futuros.

Como o resultado final do simulador levará em conta o desempenho coletivo, ou seja, o cenário como um todo, essa metodologia de cálculo se mostrou como um bom ponto de partida para analisar os impactos da posição da EB.

Espera-se que quanto mais pontos houverem na matriz, mais cálculos de perda serão realizados. Consequentemente, o tempo de espera da simulação será consideravelmente maior. Devido a isso, é necessário encontrar uma quantidade de pontos que seja mais adequada para que a simulação não seja muito demorada e a quantidade de requisições não exceda o limite da API utilizada.

Após a otimização da posição da EB, novos cálculos de perda entre a EB (posicionada no ponto otimizado) e cada usuário são realizados, e os resultados utilizados no simulador Vienna 5G SL são os obtidos a partir desta nova simulação.

Os resultados oriundos da etapa anterior até poderiam ser utilizados, mas podem não ser tão precisos devido à menor quantidade de pontos utilizados para diminuir a sobrecarga na simulação.

### **3.4. Cenários de simulação**

Neste trabalho, foram realizadas 4 simulações no Vienna 5G SL e 6 simulações no software desenvolvido para o cálculo de perda completa (caminho + difração). As simulações no Vienna foram divididas em dois cenários, A e B. Cada cenário é composto por 3 usuários e 1 EB.

No Cenário A, os usuários foram posicionados no campus Santa Mônica da Universidade Federal de Uberlândia. Neste mesmo cenário, três simulações foram realizadas. A primeira foi para determinar as perdas sem otimizar a posição da EB. Na segunda simulação, o objetivo foi determinar a melhor posição para a EB, e na terceira simulação, o objetivo foi semelhante ao da primeira, mas a posição da EB foi alterada para local determinado pela segunda simulação.

Os dois cenários foram desenvolvidos da mesma forma, mas no segundo cenário (Cenário B), o local utilizado como base para simulação foi o campus Glória da Universidade Federal de Uberlândia.

A tabela a 3.1 apresenta as principais informações do cenário criado no simulador Vienna 5G SL.

Tabela 3.1. Parâmetros principais de simulação do Vienna 5G SL.

<b>Parâmetros das antenas</b>	<b>Valor</b>
Frequência central	2 GHz
Tipo	Três Setores
Número de transmissores	4
Tecnologia	NetworkElementTechnology.NRMN_5G

Fonte: o Autor.

A tabela 3.2 mostra as informações de configurações dos usuários.

Tabela 3.2. Parâmetros principais dos usuários no cenário de simulação do Vienna 5G SL.

<b>Parâmetros dos usuários</b>	<b>Valor</b>
Posição	Estática
Número de receptores	2
Tipo de movimentação	Posição constante
Tecnologia	NetworkElementTechnology.NRMN_5G
Modelo de canal	<i>Pedestrian A</i>

Fonte: o Autor.

Em relação ao posicionamento dos usuários e da EB, as tabelas 3.3, 3.4, 3.5 e 3.6 mostram os locais para cada um, em cada cenário.

Tabela 3.3. Coordenadas dos usuários e da EB para o cenário A. Posição da EB não otimizada.

CENÁRIO A	
<b>Nome</b>	UFU – Santa Mônica – Cenário sem otimização de EB
<b>Qtd Usuários</b>	3
<b>Resultados do cenário</b>	
<b>Usuário</b>	<b>Posição – Latitude, Longitude</b>
U1	"-18.916893778916112, -48.25859329428672"
U2	"-18.917668859414686, -48.25549044328679"
U3	"-18.91868794126846, -48.259131931012874"
EB	"-18.91148799449006, -48.25774341115139"

Fonte: o Autor.

Tabela 3.4. Coordenadas dos usuários e da EB para o cenário A. Posição da EB otimizada.

CENÁRIO A	
<b>Nome</b>	UFU - Santa Mônica - Cenário com otimização de EB
<b>Qtd Usuários</b>	3
<b>Resultados do cenário</b>	
<b>Usuário</b>	<b>Posição - Latitude, Longitude</b>
U1	"-18.916893778916112, -48.25859329428672"
U2	"-18.917668859414686, -48.25549044328679"
U3	"-18.91868794126846, -48.259131931012874"
EB	"-18.9173423195042, -48.25822155908135"

Fonte: o Autor.

Tabela 3.5. Coordenadas dos usuários e da EB para o cenário B. Posição da EB não otimizada.

CENÁRIO B	
<b>Nome</b>	UFU - Glória - Cenário sem otimização de EB
<b>Qtd Usuários</b>	3
<b>Resultados do cenário</b>	
<b>Usuário</b>	<b>Posição - Latitude, Longitude</b>
U1	"-18.94724224486452, -48.21802545999803"
U2	"-18.94340064860525, -48.21600772173306"
U3	"-18.94561284548452, -48.21050382881949"
EB	"-18.946136160898366, -48.21172763322154"

Fonte: o Autor.

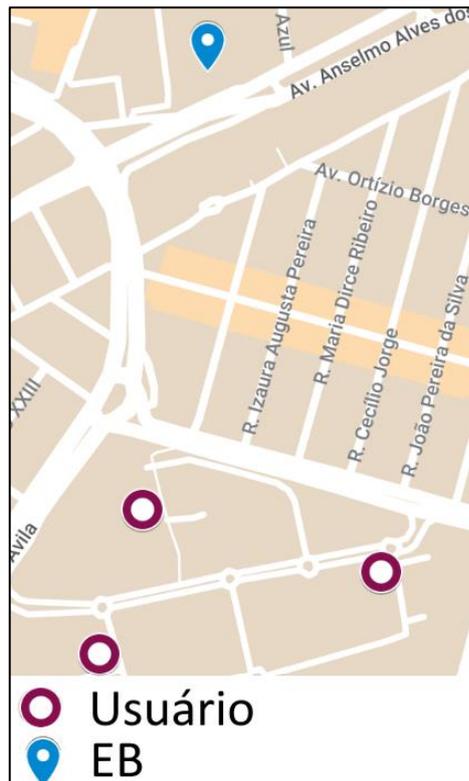
Tabela 3.6. Coordenadas dos usuários e da EB para o cenário B. Posição da EB otimizada.

CENÁRIO B	
<b>Nome</b>	UFU - Glória - Cenário com otimização de EB
<b>Qtd Usuários</b>	3
<b>Resultados do cenário</b>	
<b>Usuário</b>	<b>Posição - Latitude, Longitude</b>
U1	"-18.94724224486452, -48.21802545999803"
U2	"-18.94340064860525, -48.21600772173306"
U3	"-18.94561284548452, -48.21050382881949"
EB	"-18.944361047670068, -48.216145052203395"

Fonte: o Autor.

Para a realização dos cálculos, a altura da EB foi definida igual a 6 metros, e a altura do aparelho do usuário foi definida como 1,5 metros. Além disso, o ganho das antenas foi fixado em 1. As figuras 3.3, 3.4, 3.5 e 3.6 apresentam as posições em um mapa, para melhor compreensão do cenário.

Figura 3.3. Cenário A. Posição da EB não otimizada.



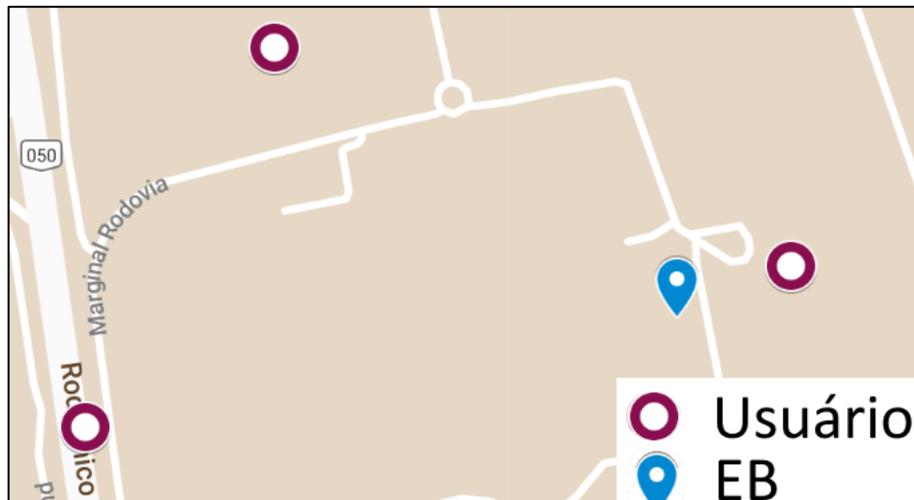
Fonte: o Autor.

Figura 3.4. Cenário A. Posição da EB otimizada.



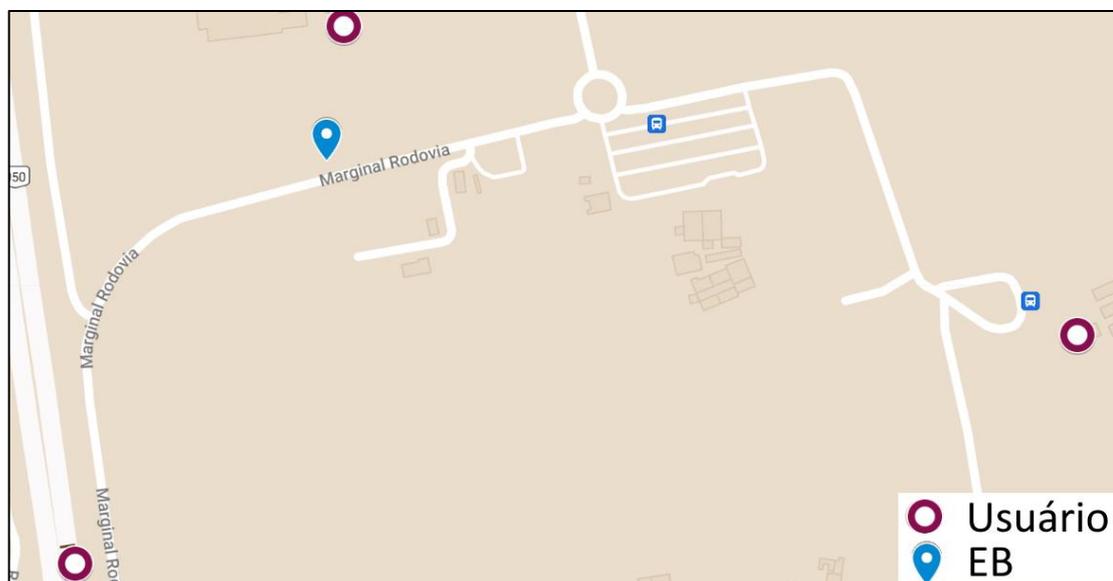
Fonte: o Autor.

Figura 3.5. Cenário B. Posição da EB não otimizada.



Fonte: o Autor.

Figura 3.6. Cenário B. Posição da EB otimizada.



Fonte: o Autor.

### 3.5. Considerações finais

Neste capítulo, a metodologia utilizada para o desenvolvimento do trabalho foi apresentada, evidenciando os cenários trabalhados e os fundamentos para a obtenção e compreensão dos resultados. Alguns conceitos foram apresentados, como o conceito de API, e uma visão geral sobre o funcionamento do software desenvolvido.

A partir das tabelas 3.3, 3.4, 3.5 e 3.6, e das figuras 3.3, 3.4, 3.5 e 3.6, algumas observações a respeito do cenário podem ser realizadas. Em primeiro lugar, como os pontos são próximos entre si, espera-se que a perda de caminho seja baixa.

O foco deste trabalho não é a perda de caminho. Por conta disso, o cenário foi selecionado de tal forma que a mesma não seja tão significativa. Como as distâncias são semelhantes, espera-se que os valores de perda de caminho sejam mais ou menos semelhantes também.

A prioridade é analisar o impacto da difração e como ela afetará a intensidade do sinal. Para a perda de caminho, considerou-se a menor perda possível, aproximando-se de uma situação real, onde os ganhos das antenas e a potência efetiva radiada (ERP – *Effective Radiated Power*) são grandes o suficiente para o receptor. Isso é importante por conta da sensibilidade as perdas que o simulador possui.

No capítulo 4 são vistos os resultados dos cálculos de perda por difração. Espera-se que em pontos próximos da EB, como na posição de alguns usuários nas figuras 3.3, 3.4 e 3.5, a perda por difração pode ser menor devido a menor variação do perfil do terreno em curta distância, ou pela compensação pela altura do transmissor, proporcionando um cenário com a linha de visada completamente desobstruída, ou pouco obstruída.

Assim, o principal objetivo é analisar os efeitos da perda por difração no desempenho de uma rede móvel em comparação com um modelo de perdas mais simples, que considera somente a perda básica com a distância. Outra medição útil deste trabalho é a comparação de desempenho entre dois cenários distintos. Em um deles, a estação base é posicionada de forma aleatória, e em outro, sua posição foi otimizada.

## **Capítulo 4 – Resultados**

### **4.1. Introdução**

Nesta seção, serão apresentados os resultados de simulação, assim como a análise destes. No capítulo 3 descreve-se o cenário de simulação, e na seção 3.5 são realizadas considerações com relação aos resultados esperados.

Para apresentação dos resultados, essa seção contará com tabelas e figuras que mostrarão os dados de saída das simulações. Na seção 4.2 são apresentados os resultados pertinentes a perda por difração, obtidos pelo software desenvolvido e, na seção 4.3 os resultados são do simulador Vienna 5G SL, para todos os cenários.

### **4.2. Resultados dos cálculos das perdas por difração**

Conforme apresentado no capítulo 3, este trabalho conta com 2 cenários de simulação. Para cada cenário, foram realizadas 3 simulações. Em uma delas, calculou-se a perda por difração em um cenário onde a posição da EB não foi otimizada.

Em outra simulação, foi determinada a melhor posição para a EB, considerando a menor perda média entre os usuários. Por fim, a última simulação foi semelhante a primeira, com a diferença que a EB foi posicionada no local determinado pela simulação anterior, o local otimizado.

Os resultados relativos ao cenário A podem ser vistos nas tabelas 4.1 e 4.2.

Tabela 4.1. Resultados dos cálculos de perda completa no cenário A. Posição da EB não otimizada.

Usuário	Perdas				
	Friis	Dois Raios	Maior perda de caminho	Difração	Completa (maior perda de caminho + perda por difração)
U1	94.1433dB	92.2629dB	94.1433dB	12.2102dB	106.3535dB
U2	95.7043dB	95.3764dB	95.7043dB	13.7597dB	109.4641dB
U3	96.6795dB	97.3361dB	97.3361dB	8.4197dB	105.7558dB

Fonte: o Autor.

Tabela 4.2. Resultados dos cálculos de perda completa no cenário A. Posição da EB otimizada.

Usuário	Perdas				
	Friis	Dois Raios	Maior perda de caminho	Difração	Completa (maior perda de caminho + perda por difração)
U1	74.7285dB	52.9923dB	74.7285dB	0	74.7285dB
U2	87.7035dB	79.3852dB	87.7035dB	1.5767dB	89.2802dB
U3	83.4883dB	70.8974dB	83.4883dB	0	83.4883dB

Fonte: o Autor.

De acordo com as tabelas 4.1 e 4.2, é possível observar que a metodologia empregada para otimizar a posição da EB apresentou bons resultados. A perda completa para cada usuário apresentada na tabela 4.2 possui uma média menor quando comparadas ao valor médio das perdas completas apresentadas na tabela 4.1.

Após a otimização da posição da EB, houve uma redução significativa da perda por difração e da perda de caminho, mesmo considerando o maior valor entre os dois valores de perda de caminho calculados.

Por conta da redução do valor médio da perda, espera-se que o desempenho da rede no cenário da EB com a posição otimizada seja superior ao cenário com a EB posicionada em um ponto não ideal. Abaixo, podem ser vistos os resultados referentes ao cenário B, nas tabelas 4.3 e 4.4.

Tabela 4.3. Resultado dos cálculos de perda completa no cenário B. Posição da EB não otimizada.

Usuário	Perdas				
	Friis	Dois Raios	Maior perda de caminho	Difração	Completa (maior perda de caminho + perda por difração)
U1	95.0382dB	94.0530dB	95.0382dB	0.9514dB	95.9896dB
U2	93.1689dB	90.3161dB	93.1689dB	0.9977dB	94.1666dB
U3	81.4687dB	66.9149dB	81.4687dB	1.5653dB	83.0340dB

Fonte: o Autor.

Tabela 4.4. Resultado dos cálculos de perda completa no cenário B. Posição da EB otimizada.

Usuário	Perdas				
	Friis	Dois Raios	Maior perda de caminho	Difração	Completa (maior perda de caminho + perda por difração)
U1	89.9892dB	83.9456dB	89.9892dB	0	89.9892dB
U2	79.1187dB	62.2141dB	79.1187dB	0	79.1187dB
U3	94.1666dB	92.3114dB	94.1666dB	0	94.1666dB

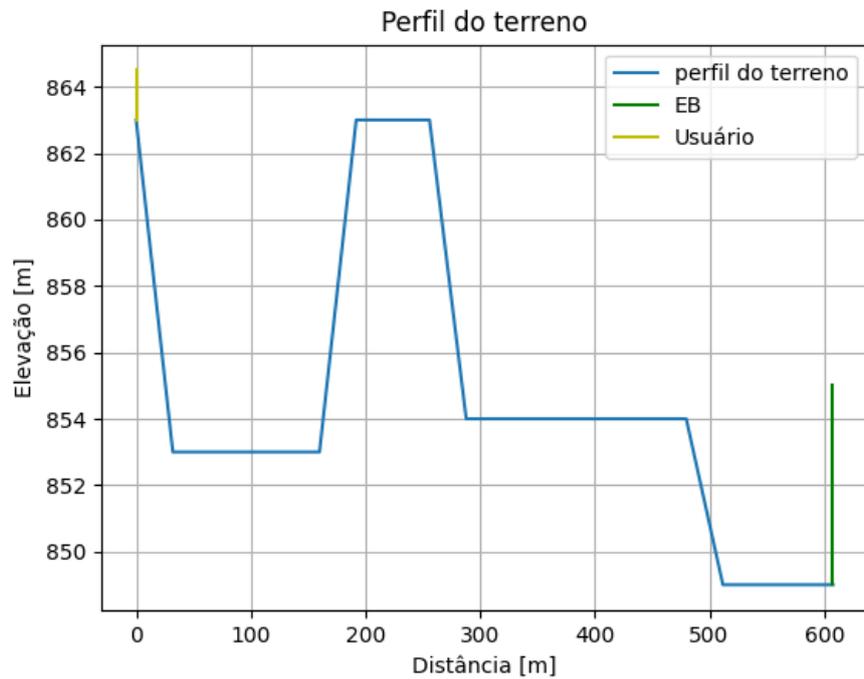
Fonte: o Autor.

De maneira semelhante ao que foi visto no cenário A, os resultados apresentados no cenário B mostram com clareza a diferença no valor médio da perda quando a posição da EB é otimizada.

No cenário B, nota-se que o valor da perda por difração é igual a zero para todos os usuários, indicando um caso de visada direta entre os usuários e a EB.

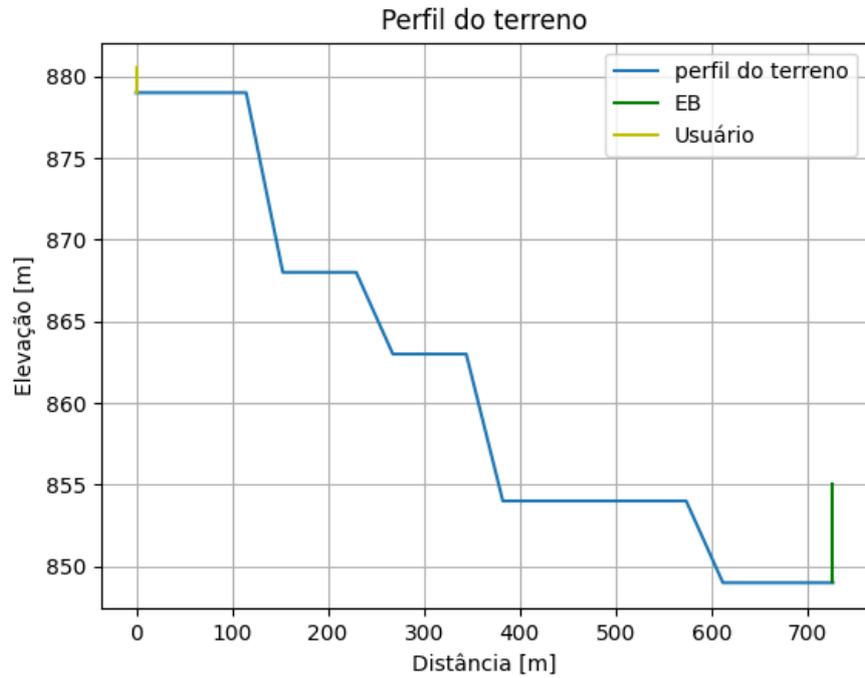
As figuras 4.1 a 4.12 mostram o perfil do terreno entre cada usuário e a EB, fornecendo uma informação útil para a análise dos resultados apresentados nas tabelas 4.1 a 4.4.

Figura 4.1. Perfil do terreno entre U1 e a EB no cenário A. Posição da EB não otimizada.



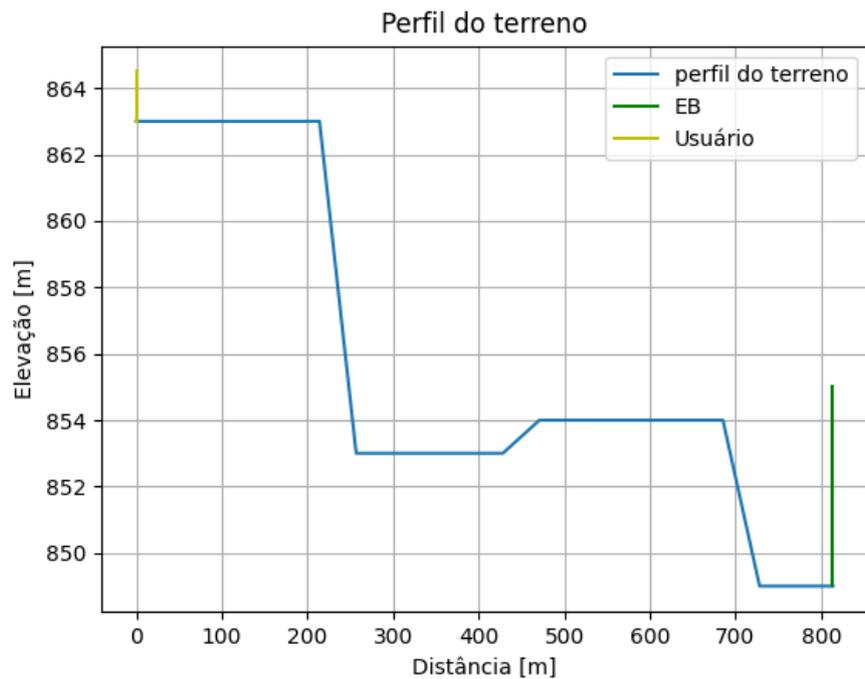
Fonte: o Autor.

Figura 4.2. Perfil do terreno entre U2 e a EB no cenário A. Posição da EB não otimizada.



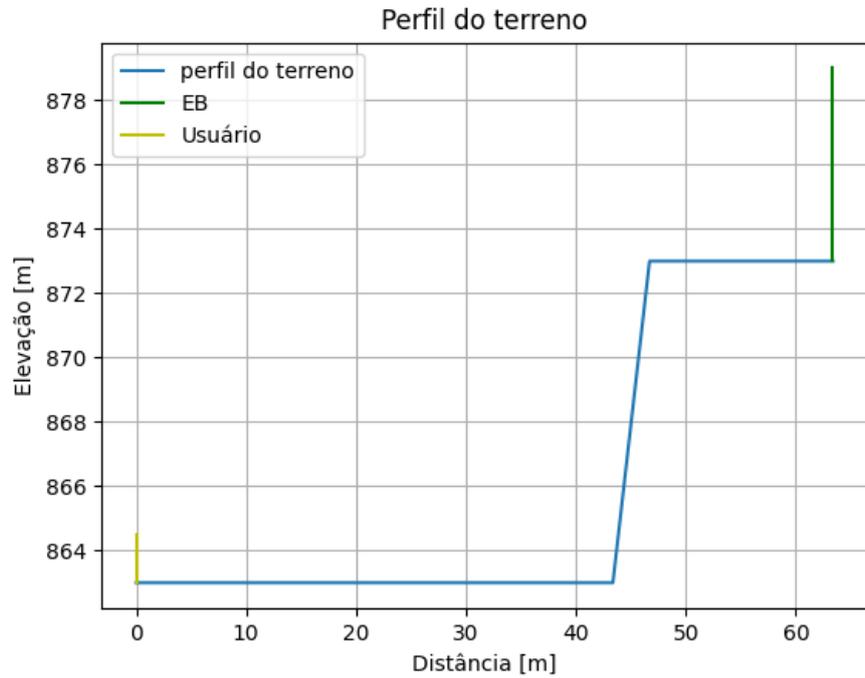
Fonte: o Autor.

Figura 4.3. Perfil do terreno entre U3 e a EB no cenário A. Posição da EB não otimizada.



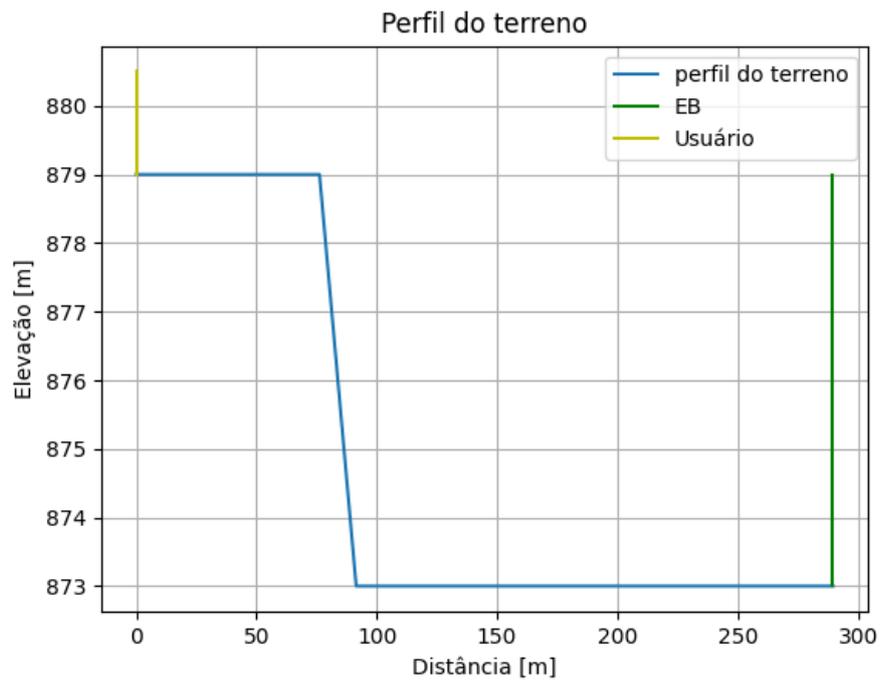
Fonte: o Autor

Figura 4.4. Perfil do terreno entre U1 e a EB no cenário A. Posição da EB otimizada.



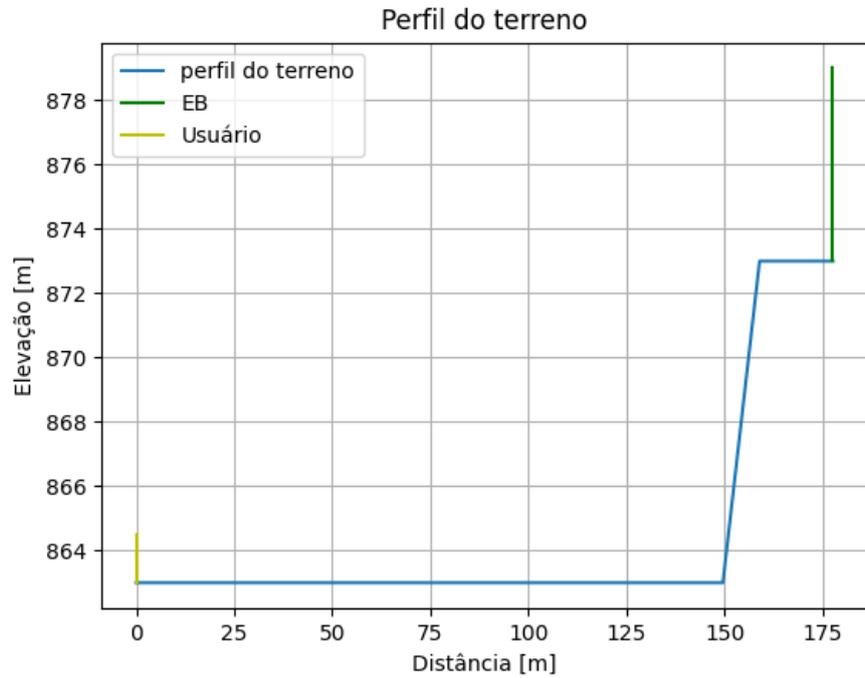
Fonte: o Autor.

Figura 4.5. Perfil do terreno entre U2 e a EB no cenário A. Posição da EB otimizada.



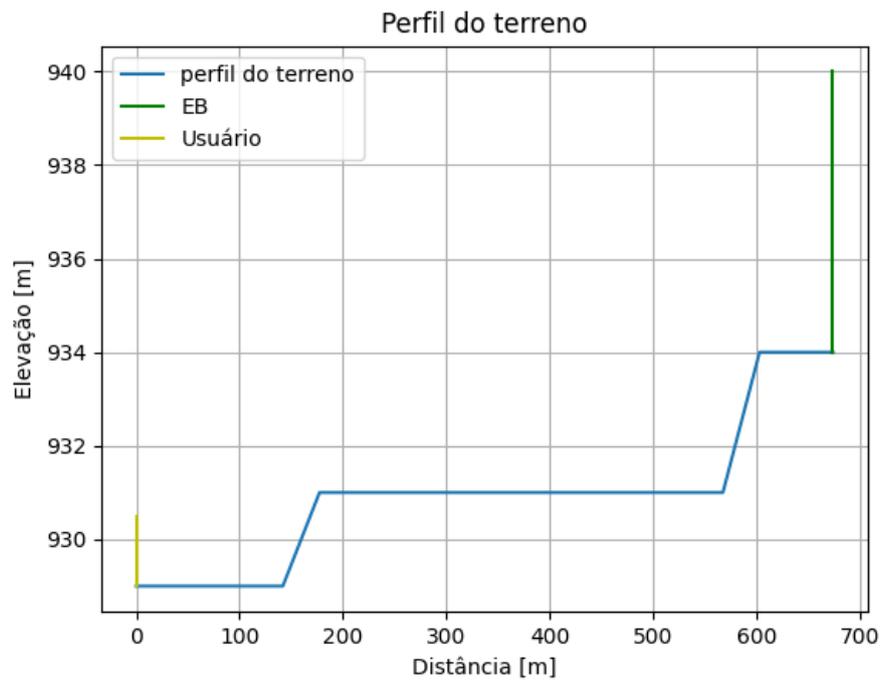
Fonte: o Autor.

Figura 4.6. Perfil do terreno entre U3 e a EB no cenário A. Posição da EB otimizada.



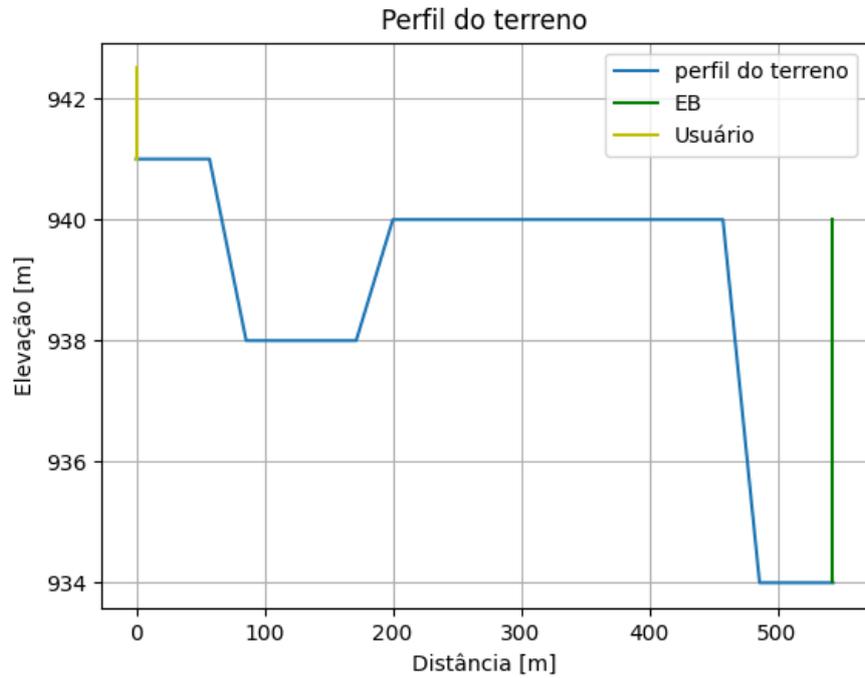
Fonte: o Autor.

Figura 4.7. Perfil do terreno entre U1 e a EB no cenário B. Posição da EB não otimizada.



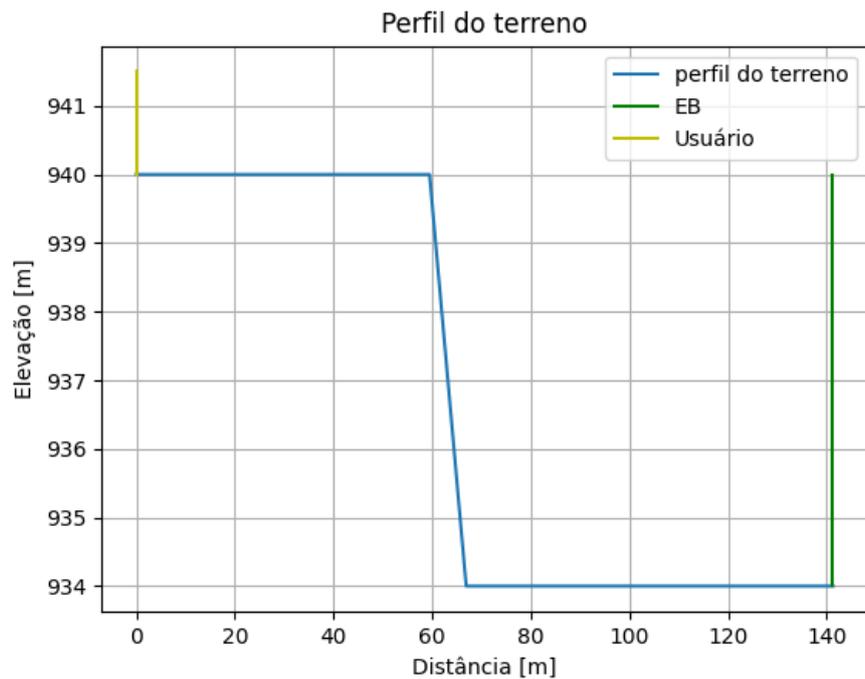
Fonte: o Autor.

Figura 4.8. Perfil do terreno entre U2 e a EB no cenário B. Posição da EB não otimizada.



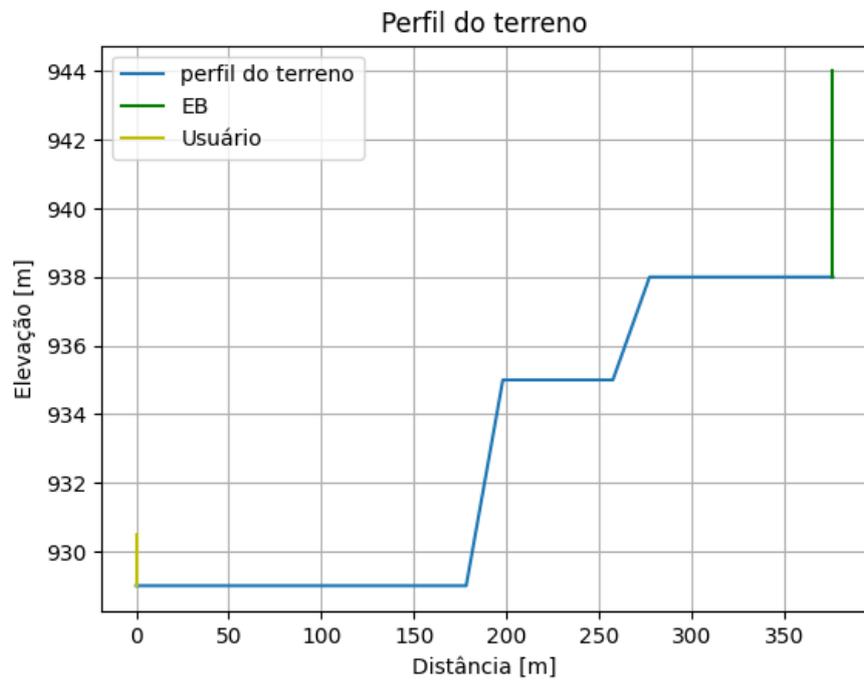
Fonte: o Autor.

Figura 4.9. Perfil do terreno entre U3 e a EB no cenário B. Posição da EB não otimizada.



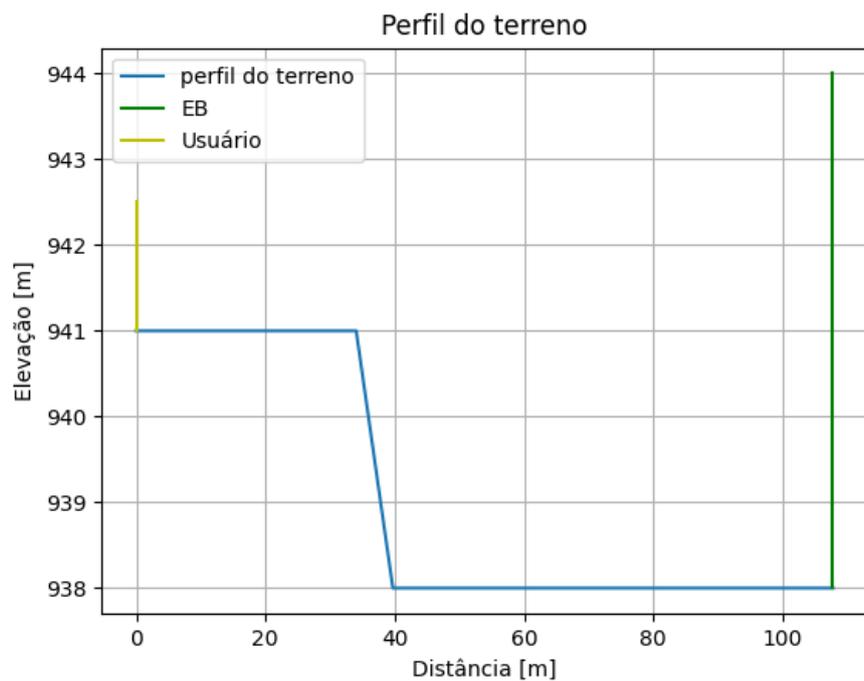
Fonte: o Autor.

Figura 4.10. Perfil do terreno entre U1 e a EB no cenário B. Posição da EB otimizada.



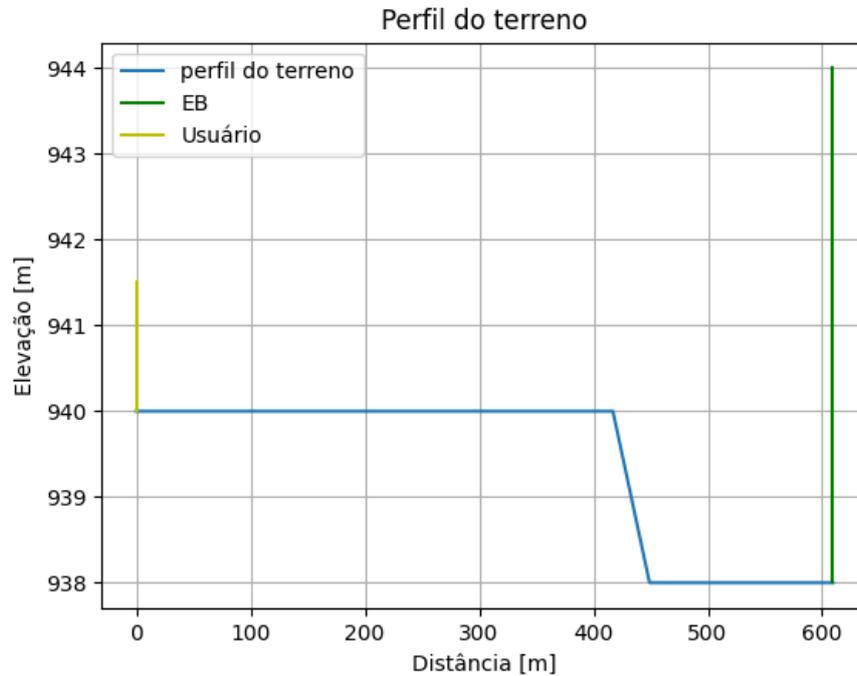
Fonte: o Autor.

Figura 4.11. Perfil do terreno entre U2 e a EB no cenário B. Posição da EB otimizada.



Fonte: o Autor.

Figura 4.12. Perfil do terreno entre U3 e a EB no cenário B. Posição da EB otimizada.



Fonte: o Autor.

Em um cenário real, considerando células maiores, a redução pode ser ainda mais significativa, aumentando a eficiência de transmissão dos sistemas de comunicações móveis.

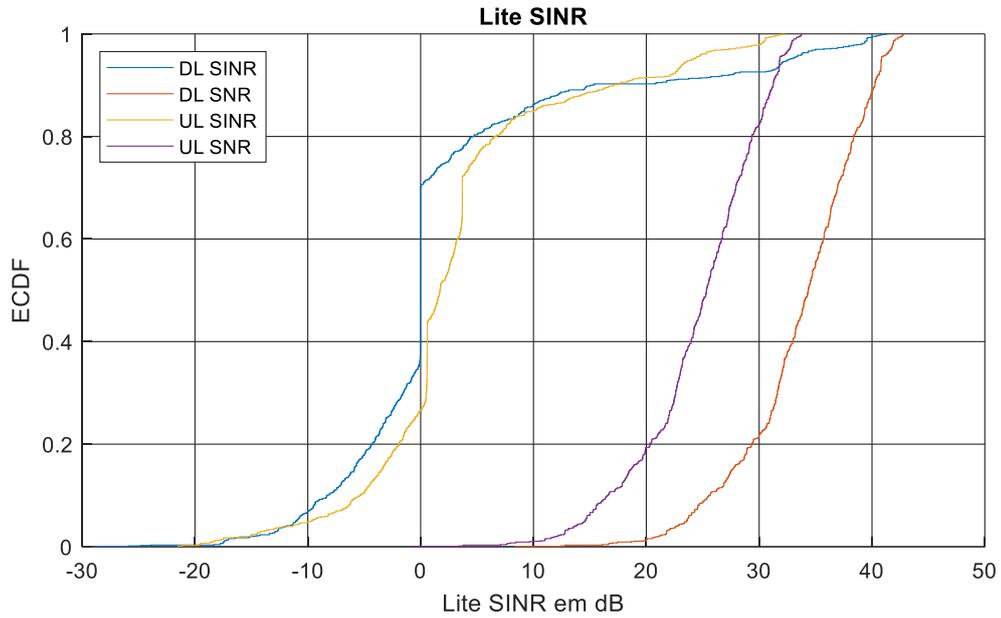
### 4.3. Resultados de simulação

Antes de analisar os resultados, é importante saber sobre os dois algoritmos de escalonamento implementados no simulador. Um deles é o Round Robin (RR), com o nome DLFeedback nos resultados de simulação, e o outro é o de melhor indicador de qualidade do canal (BestCQI – *Best Channel Quality Indicator*).

O RR é um algoritmo que serve o primeiro pacote que chega (FCFS – *first come first served*) e divide as linhas de trabalho e utiliza com uma unidade de tempo chamada *time quantum* ou *time slice*, que varia de 10 a 100 milissegundos [14]. Em cada unidade de tempo, uma fração da linha de trabalho em questão é realizada, tudo isso de maneira circular.



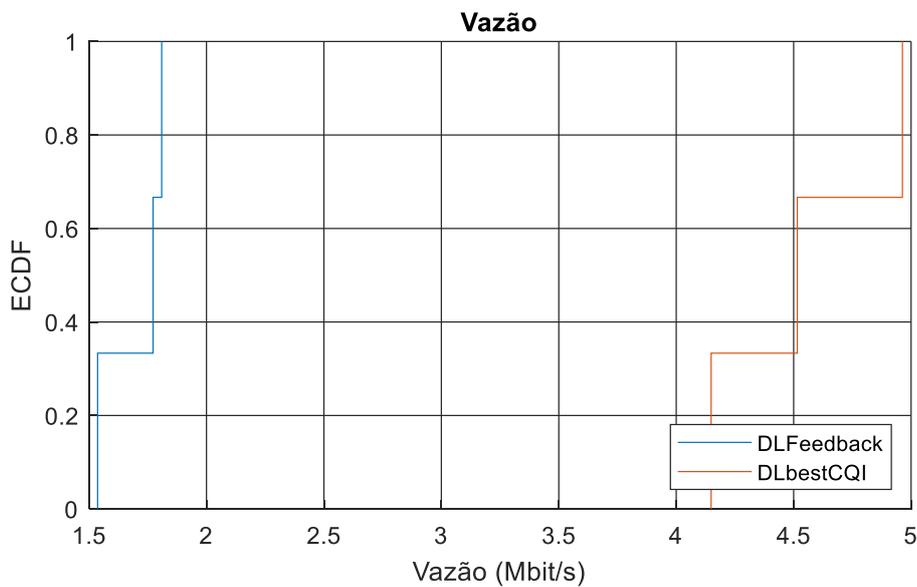
Figura 4.14. Resultado: SINR para o cenário A. Posição da EB não otimizada.



Fonte: o Autor.

A partir da figura 4.13, fica evidente a diferença de desempenho entre o algoritmo de escalonamento BestCQI, que considera indicativos de qualidade do canal para fazer a alocação de recursos, e o RR, que é de maior simplicidade.

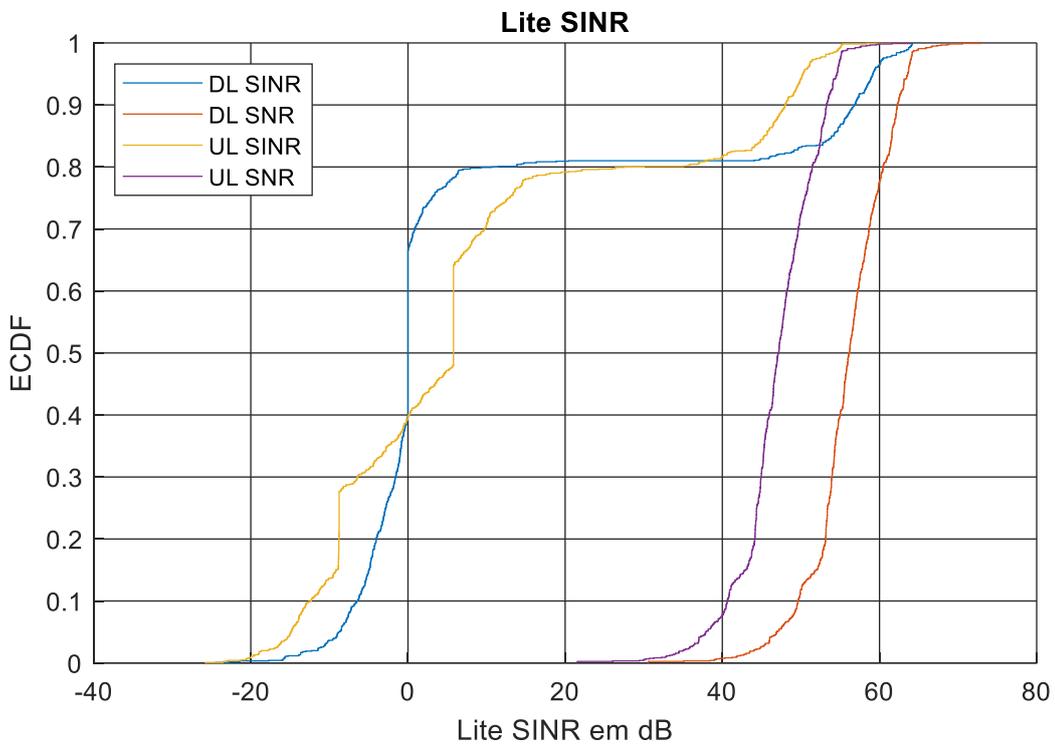
Figura 4.15. Resultado: Vazão para o cenário A. Posição da EB otimizada.



Fonte: o Autor.

Comparando os resultados de vazão no cenário A, a figura 4.13 indica uma vazão máxima em torno de 4,5Mbit/s para o algoritmo de escalonamento BestCQI e uma vazão máxima de aproximadamente 1,76Mbit/s para o algoritmo de escalonamento RR. Por outro lado, após a otimização da posição da EB, os resultados de vazão apresentados na figura 4.15 indicam uma taxa máxima igual a aproximadamente 5Mbit/s, evidenciando uma melhora nos resultados.

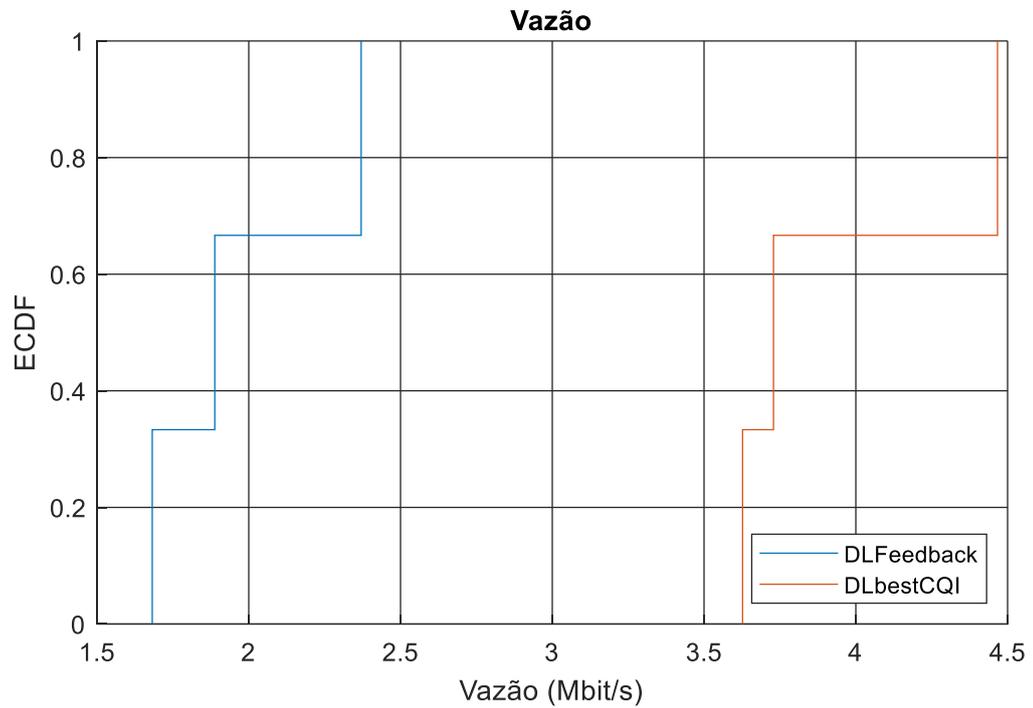
Figura 4.16. Resultado: SINR para o cenário A. Posição da EB otimizada.



Fonte: o Autor.

As medidas de SINR apresentadas na figura 4.16 apresentam valores máximos mais elevados que os apresentados na figura 4.14. Uma justificativa para isso é o uso de três setores no sistema e a posição da EB em relação aos usuários, melhorando a SINR.

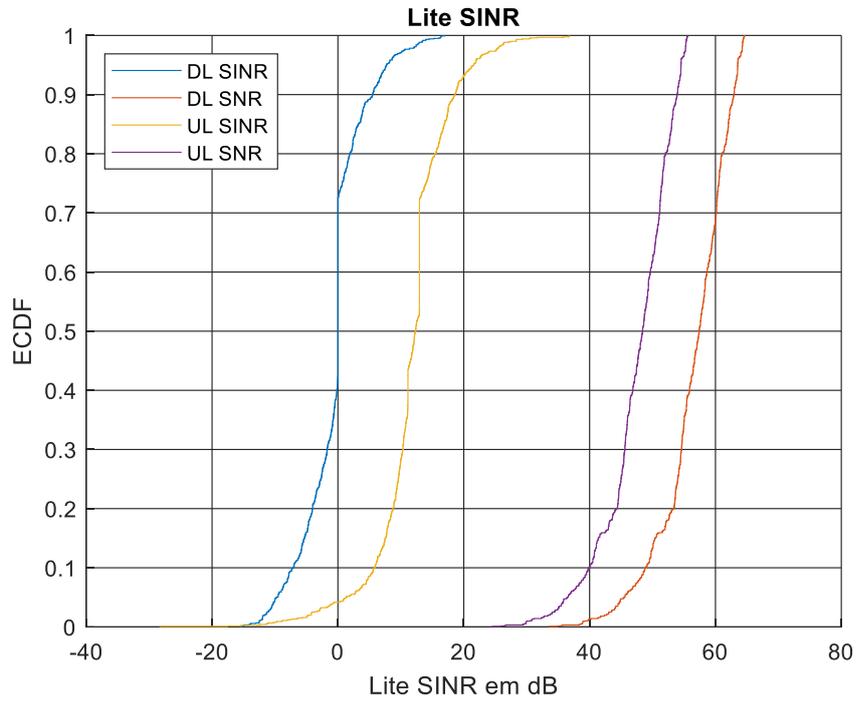
Figura 4.17. Resultado: Vazão para o cenário B. Posição da EB não otimizada.



Fonte: o Autor.

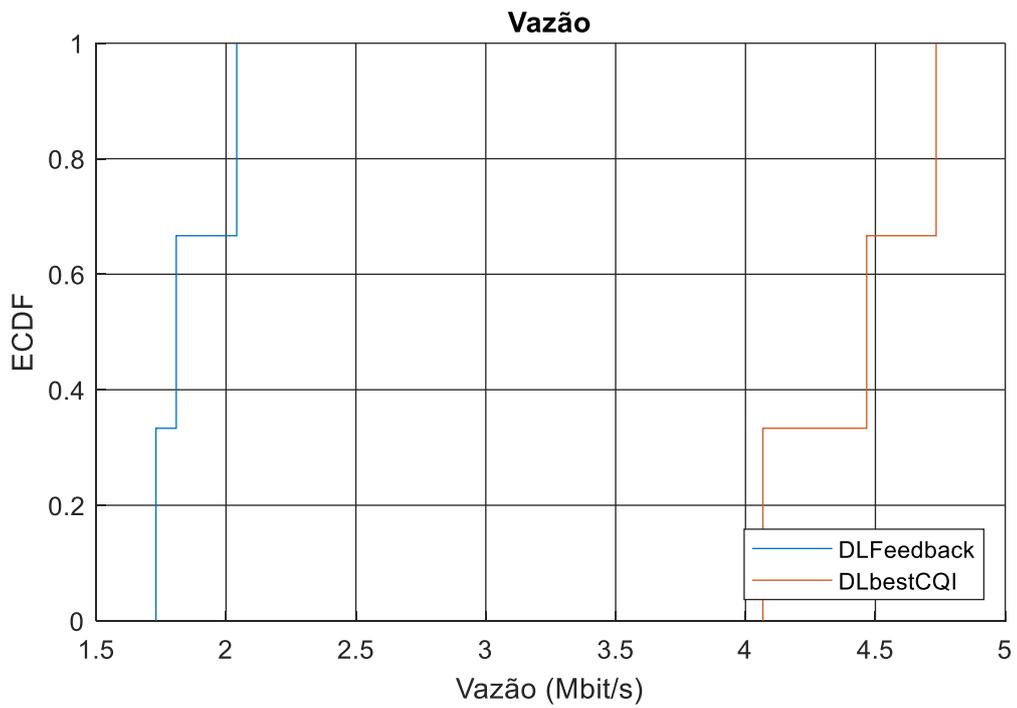
Assim como a figura 4.13 e 4.15, a figura 4.17 mostra diferenças de desempenho significativas entre os algoritmos de escalonamento RR e BestCQI.

Figura 4.18. Resultado: SINR para o cenário B. Posição da EB não otimizada.



Fonte: o Autor.

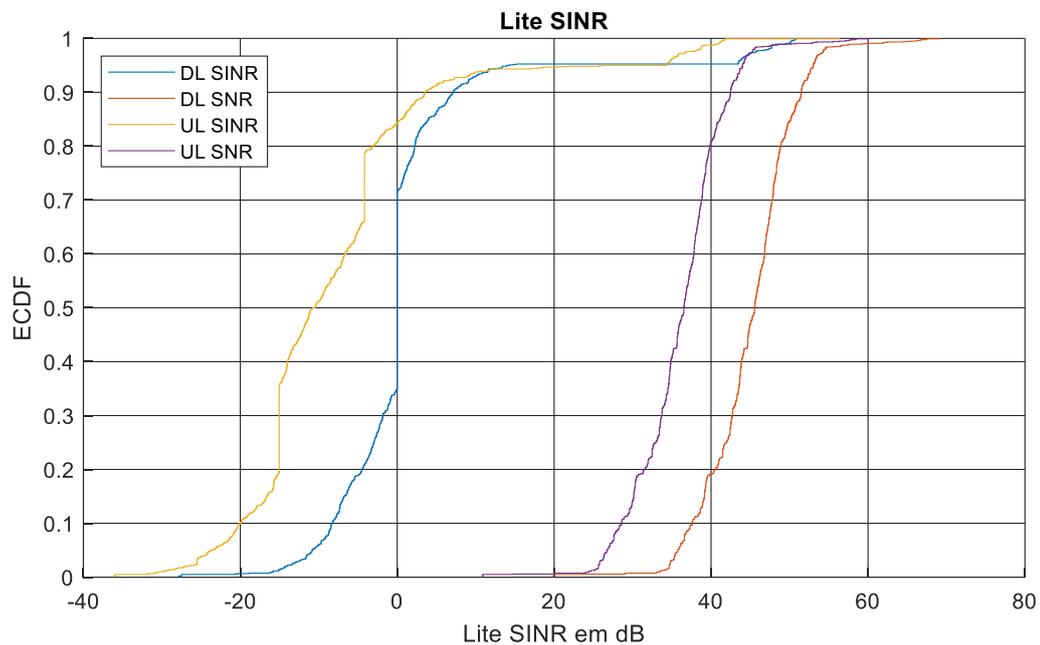
Figura 4.19. Vazão para o cenário B. Posição da EB otimizada.



Fonte: o Autor.

A figura 4.19 indica uma vazão máxima em torno de 4,75Mbit/s para o algoritmo de escalonamento BestCQI. Esse valor é maior que o valor da vazão máxima apresentado na figura 4.17. Além disso, a vazão mínima apresentada na figura 4.19 também é superior a vazão mínima apresentada na figura 4.17. Isso certamente ocorreu por conta do menor valor médio da perda após a otimização da posição da EB.

Figura 4.20. SINR para o cenário B. Posição da EB otimizada.



Fonte: o Autor.

Em relação a SINR, os resultados apresentados pela figura 4.20 indicam uma melhora nos valores máximos da SINR quando comparados aos apresentados na figura 4.18. A geometria apresentada na figura 3.4 indica que a EB provavelmente não está utilizando os três setores e, por conta disso, percebe-se um aumento de interferência para este caso.

#### **4.4. Considerações finais**

No presente capítulo, foram apresentados os resultados dos cálculos de perda completa (difração e caminho), destacando-se a perda por difração, que não é levada em conta pelo simulador. Não foram realizadas simulações considerando apenas a perda de caminho porque para alguns usuários nos cenários com a posição da estação-base otimizada, o valor da perda por difração foi igual a zero, restando apenas a perda de caminho.

A partir dos resultados das simulações, foi possível observar quais foram as consequências da difração no resultado final, e como o posicionamento da EB pode influenciar no desempenho da rede.

Como o objetivo do algoritmo de otimização da posição da EB foi posicioná-la no local onde a perda foi a menor possível, o algoritmo escolheu o ponto onde a perda média entre cada usuário foi menor. Em alguns casos, não foi observada a obstrução da linha de visada.

## Capítulo 5 – Conclusão

As redes 5G tem por objetivo melhorar a qualidade e a experiência dos usuários, oferecendo um serviço mais rápido e uma experiência mais fluida. Para que esse objetivo seja alcançado, é necessário conhecimento a respeito dos fenômenos físicos relacionados ao eletromagnetismo e a propagação. Um dos fenômenos que interferem na propagação das ondas eletromagnéticas é a difração, que tem sido objeto de estudo para muitos pesquisadores.

A implementação de redes 5G reais para a realização de testes não é uma opção vantajosa, sobretudo por conta do alto custo dos equipamentos e da dificuldade em projetar cenários realistas por conta de toda a infraestrutura necessária. Para solucionar esse problema, diversos simuladores de redes foram desenvolvidos, sendo o NS3 e o Vienna 5G SL entre os mais usados até o momento. Todavia, boa parte desses simuladores, como o Vienna 5G SL, não consideram a perda por difração em seus cálculos.

Neste trabalho, dois cenários foram simulados. No cenário A, os usuários foram posicionados em uma distância maior entre si em comparação com o cenário B, e a EB foi posicionada em um ponto ainda mais distante antes da otimização de sua posição.

A partir dos resultados apresentados no capítulo 4, também foi possível observar quais os impactos da difração em um cenário de redes móveis. Essas observações contribuem para uma análise mais realista do desempenho das redes simuladas, ainda que os cenários modelados neste trabalho sejam simplificados em relação a um cenário real.

O modelo utilizado para o cálculo de perda por difração não é o mais preciso e seus resultados tendem a ser otimistas em relação à perda [1]. Por conta disso, um possível trabalho futuro consiste na aplicação de um modelo que entrega resultados mais precisos sobre a perda de caminho e difração, contribuindo para um resultado mais realista.

Em todos os casos, conclui-se que é necessário determinar a posição da EB levando em conta não apenas a perda média e a difração, mas também as fontes de interferência. Em um sistema com muitos usuários e muitos clusters, a interferência pode causar impactos significativos que atuarão limitando a capacidade do sistema.

Espera-se que com o uso das antenas inteligentes, possa ocorrer um aumento tanto da SNR, por conta da maior diretividade das antenas, quanto da SINR, assim como ocorre em

sistemas setorizados. Todavia, em um sistema muito dinâmico e com muitos usuários, em condições pessimistas, também pode ocorrer uma redução da SINR para um usuário em particular.

Outras melhorias que podem ser implementadas em trabalhos futuros são novas propostas de otimização da posição da EB, levando em conta não somente a perda de caminho, mas também a degradação gerada pela baixa SINR.

Por fim, mesmo utilizando um modelo simples para o cálculo de perda por difração, o objetivo deste trabalho foi alcançado, evidenciando as consequências que a difração traz em um sistema de redes móveis e como o posicionamento da EB pode impactar nas medidas de perda, vazão, SNR e SINR.

## Referências

- [1]. RAPPAPORT, Theodore S. Comunicações sem fio: princípios e práticas. 2. ed. São Paulo
- [2]. STUTZMAN, Warren L. Antenna theory and design. 3rd. ed. Hoboken: Wiley, 2013. 822p
- [3]. BALANIS, Constantine A. Teoria de antenas: análise e síntese. Rio de Janeiro: Livros Técnicos e Científicos, 2009. Vol. 1.
- [4]. LATHI, B. P. Sistemas de comunicação. Rio de Janeiro: Guanabara, 1987, c1979. 401p.
- [5]. MÜLLER, Martin K. et. al. Flexible multi-node simulation of cellular mobile communications: the Vienna 5G System Level Simulator. Viena: EURASIP Journal on Wireless Communications and Networking, 2018.
- [6]. [https://www.3gpp.org/ftp/tsg\\_ran/tsg\\_ran/tsg\\_r\\_16/docs/PDF/RP-020376.pdf](https://www.3gpp.org/ftp/tsg_ran/tsg_ran/tsg_r_16/docs/PDF/RP-020376.pdf). Acesso 20/03/2024
- [7]. <https://www.geeksforgeeks.org/5g-network-architecture/>. Acesso 20/03/2024
- [8]. <https://open-elevation.com/> Acesso 20/03/2024
- [9]. <https://developers.google.com/maps/documentation/elevation/overview?hl=pt-br> Acesso 20/03/2024
- [10]. KUROSE, James F. Redes de computadores e a Internet: uma abordagem top-down. São Paulo: Pearson Education do Brasil, 2013. 634 p.
- [11]. Google Cloud Platform: <https://console.cloud.google.com/welcome?hl=pt-br&pli=1&project=dynamic-tide-402302>. Acesso 20/03/2024
- [12]. DOLUKHANOV, M. Propagation of Radio Waves. Moscow: URSS, 1995. 372 p.
- [13]. <https://www.openstreetmap.org/#map=9/-19.9283/-45.0385>. Acesso 20/03/2024
- [14]. [https://medium.com/@francescofranco\\_39234/round-robin-scheduling-algorithm-8d65128d96db](https://medium.com/@francescofranco_39234/round-robin-scheduling-algorithm-8d65128d96db). Acesso 20/03/2024

## Apêndice A – Código do software para cálculo de perda

Arquivo: **apis.py**

```
from math import *
from decouple import config

def extract_lat_lon(lop):
    #lop = list of points. Format: lop = [[lat0,lon0],[lat1,lon1]...]
    lat,lon = [],[]
    for l in lop:
        lat.append(l[0])
        lon.append(l[1])
    return lat,lon

def free_elevation_api(lt0, ln0, lt1, ln1, samples):
    lat0 = float(lt0)
    lon0 = float(ln0)
    lat1 = float(lt1)
    lon1 = float(ln1)
    a = (lon1-lon0)/(lat1-lat0)
    step = (lat1-lat0)/(samples-1)
    lat = [lat0+i*step for i in range(samples)]
    lon = [lon0 + a*(lat[i] - lat0) for i in range(samples)]
    data = [f'{lat[i]},{lon[i]}|' for i in range(samples)]
    url = f'https://api.open-
elevation.com/api/v1/lookup?locations={''.join(data)}'
    #print(url)
    return url

def google_elevation_api(lat0, lon0, lat1, lon1, samples):
    return
f'''https://maps.googleapis.com/maps/api/elevation/json?path={lat0}%2C{lon0}%7
C{lat1}%2C{lon1}&samples={samples}&key={config('GOOGLE_KEY')}'''

def two_rays_path_loss(d, htx, hrx, gtx=1, grx=1):
    #print(f'd: {d}, htx:{htx},hrx{hrx}')
    return 10*log10((d[-1]**4)/(gtx*grx*(htx**2)*(hrx**2)))

def friis_path_loss(p, d, wavel, htx, hrx, gtx=1, grx=1):
    return 10*log10((4*pi*sqrt((abs(p[0]+htx-p[-1]-hrx)**2 + d[-1]**2)))**2 /
(gtx*grx*wavel**2))
```

```

def calculate_distance(lat0, lon0, lat1, lon1):
    lat0 = float(lat0) * pi / 180
    lon0 = float(lon0) * pi / 180
    lat1 = float(lat1) * pi / 180
    lon1 = float(lon1) * pi / 180
    return acos(cos(lat0) * cos(lon0) * cos(lat1) * cos(lon1) +
cos(lat0)*sin(lon0)*cos(lat1)*sin(lon1) + sin(lat0) * sin(lat1)) * 6371e3

```

Arquivo: **diffraction\_loss\_path.py**

```

import requests
import apis
import matplotlib.pyplot as plt
from math import *

# funcao para obter o perfil do terreno
def obtain_terrain_profile(p0, p1, n=20, selection='free'):
    lat0, lon0 = p0
    lat1, lon1 = p1

    match selection:
        case 'google':
            r = requests.get(apis.google_elevation_api(lat0, lon0, lat1, lon1,
n))
        case _:
            r = requests.get(apis.free_elevation_api(lat0, lon0, lat1, lon1,
n))

    elevation = [float(el['elevation']) for el in r.json()['results']]
    distance = apis.calculate_distance(lat0, lon0, lat1, lon1)
    step = distance/(n-1)
    distances = [i*step for i in range(n)]

    return distances, elevation

def one_knife_edge_loss(h1, hobs, h2, dobs, d2, wavel, e1, e2):
    gain = 0
    beta = atan((hobs-h1-e1)/dobs)
    gama = atan((hobs-h2-e2)/(d2-dobs))
    alpha = beta + gama
    try:
        v = alpha*sqrt(2*dobs*(d2-dobs)/(wavel*d2))
        if -1<=v<=0:
            gain = 20*log10(0.5-0.62*v)
        if 0<=v<=1:
            gain = 20*log10(.5*exp(-0.95*v))

```

```

        if 1<=v<=2.4:
            gain = 20*log10(0.4-sqrt(0.1184-(0.38-0.1*v)**2))
        if v>=2.4:
            gain = 20*log10(0.225/v)
    except:
        pass
    return gain if gain < 0 else 0

def dois_maiores(indices, h, d, h1, h2):
    mds = []
    mes = []
    mds_aux = []
    mes_aux = []

    for i in indices:
        me = (h[i]-(h[0]+h1))/(d[i]-d[0])
        md = -(h[len(h)-1]+h2-h[i])/(d[len(d)-1]-d[i])
        mds.append(md)
        mds_aux.append(md)
        mes.append(me)
        mes_aux.append(me)

    mds_aux.sort()
    mes_aux.sort()

    indices_corrigeidos = []

    for i, imd in enumerate(mds):
        if mds_aux[0] == imd:
            indices_corrigeidos.append(indices[i])
            break

    for i, ies in enumerate(mes):
        if mes_aux[0] == ies:
            indices_corrigeidos.append(indices[i])
            break

    indices_corrigeidos.sort()

    return indices_corrigeidos

def calculate_equivalent_obstacle(indiceseq, d, h, h1, h2):
    indices_corrigeidos = dois_maiores(indiceseq,h,d,h1,h2) if len(indiceseq) >
2 else indiceseq

```

```

    m1 = (h[indices_corrigidos[0]]-h[0]-h1)/(d[indices_corrigidos[0]]-d[0])
    m2 = (h[len(h)-1]+h1-h[indices_corrigidos[1]])/(d[len(d)-1]-
d[indices_corrigidos[1]])
    deq = (h[indices_corrigidos[1]]-m2*d[indices_corrigidos[1]]-h1-
h[0]+m1*d[0])/(m1-m2)
    heq = h[0] + h1+m1*(deq-d[0])
    return deq, heq

```

```

def calculate_equivalent_obstacle_from_others(dob1, dob2, hob1, hob2, ht1,
ht2, h0, h1, d0, d1):
    m1 = (hob1-h0-ht1)/(dob1-d0)
    m2 = (h1+ht2-hob2)/(d1-dob2)

    d_final = (h0+ht1-hob2-m1*d0+m2*dob2)/(m2-m1)
    h_final = h0+ht1+m1*(d_final-d0)

    return d_final, h_final

```

# funcao para calcular a perda por difracao

```

def calculate_diffraction_loss(p, d, htx, wavel, hrx):
    try:
        h1 = htx #torre 1 (pode ser o tx ou rx, dependendo do perfil do
terreno)
        h2 = hrx #torre 2 (pode ser o rx ou tx, dependendo do perfil do
terreno)

        if (p[0]+htx) > (p[len(p)-1]+hrx):
            p = p[::-1]
            h1 = hrx
            h2 = htx
        # carregar theta

        debug(f'h1: {h1}. h2 {h2}. p: {p}')
        theta = -atan((p[0]+h1-p[-1]-h2)/d[-1])
        # carregar linha de visada

        debug(f'theta: {theta}')
        los = [p[0]+h1+di*tan(theta) for di in d]
        debug(f'los: {los}')
        # carregar 1 zona de Fresnel

        pzf = [(los[i]-cos(theta)*sqrt(wavel*di*(d[-1]-di))/(d[-
1]*cos(theta)))] for i, di in enumerate(d)]
        debug(f'pzf: {pzf}')
        gumes = [] # posicao dos gumes

```

```

for i in range(len(pzf)):
    if (pzf[i]-p[i] < 0):
        gumes.append(i)
#calcular_perda_final

debug(f'gumes: {gumes}')
#calc_obs
hgumes = [p[i] for i in gumes]
maximo = max(hgumes)

for a,b in enumerate(hgumes):
    if b == maximo:
        indice = gumes[a]
        break

debug(f'hgumes: {hgumes}\nmaximo:{maximo}\nindice: {indice}')
me = (maximo-(p[0]+h1))/(d[indice]-d[0])
posicoes_maior_igual_esquerdo = []

for i,j in enumerate(gumes):
    if j==indice:
        stop = i
    if j==indice+1:
        break
    he = (p[0]+h1) + me*(d[j]-d[0])
    if(p[j]>=he):
        posicoes_maior_igual_esquerdo.append(j)

md = (p[-1]+h2-maximo)/(d[-1]-d[indice])
posicoes_maior_igual_direito = []
debug(f'\nstop: {stop}\nmd {md}')

for i,j in enumerate(gumes):
    if i >= stop:
        he = maximo + md*(d[j]-d[indice])
        if (p[j] >= he):
            posicoes_maior_igual_direito.append(j)

debug(f'\nLen pos_maior_igual_direito:
{len(posicoes_maior_igual_direito)}')

perda = 0

```

```

    debug(f'\nPosicoes >= direito {posicoes_maior_igual_esquerdo}')

    if len(posicoes_maior_igual_direito) == 1 and
len(posicoes_maior_igual_esquerdo) == 1:
        perda = one_knife_edge_loss(p[0], maximo, p[len(p)-1], d[indice],
d[len(d)-1], wavel, h1, h2)

        elif len(posicoes_maior_igual_direito) > 1 and
len(posicoes_maior_igual_esquerdo) == 1:
            d_obs, h_obs =
calculate_equivalent_obstacle(posicoes_maior_igual_direito, d, p, h1, h2)
            perda = one_knife_edge_loss(p[0],h_obs, p[-1], d_obs, d[len(d)-1],
wavel, h1, h2)

        elif len(posicoes_maior_igual_direito) == 1 and
len(posicoes_maior_igual_esquerdo) > 1:
            d_obs, h_obs =
calculate_equivalent_obstacle(posicoes_maior_igual_esquerdo, d, p, h1, h2)
            perda = one_knife_edge_loss(p[0], h_obs, p[len(p)-1], d_obs,
d[len(d)-1], wavel, h1, h2)

        else:
            d_obs1, h_obs1 =
calculate_equivalent_obstacle(posicoes_maior_igual_direito, d, p, h1, h2)
            d_obs2, h_obs2 =
calculate_equivalent_obstacle(posicoes_maior_igual_esquerdo, d, p, h1, h2)
            d_obs, h_obs = calculate_equivalent_obstacle_from_others(d_obs2,
d_obs1, h_obs2, h_obs1, h1, h2, p[0], p[len(p)-1],\
                                                                    d[0],
d[len(d)-1])
            perda = one_knife_edge_loss(p[0], h_obs, p[len(p)-1], d_obs,
d[len(d)-1], wavel, h1, h2)
        except:
            return 0
        return -perda

def calculate_complete_loss(h,d, htx, wavel, hrx):
    diffraction_loss = calculate_diffraction_loss(h,d,htx,wavel,hrx)
    path_loss = calculate_path_loss(h,d,wavel,htx,hrx)
    return path_loss + diffraction_loss

# funcao para calcular a perda do caminho (perda completa, considerando a
perda por difracao)
def calculate_path_loss(p, d, wavel, htx, hrx):
    # uses the calculate_diffraction_loss function + default distance path
loss

```

```
    return max([apis.two_rays_path_loss(d, htx, hrx), apis.friis_path_loss(p,  
d, wavel, htx, hrx)])
```

```
def debug(m):  
    if debug_on:  
        print(m)
```

## Apêndice B – Código do software para otimização de estação-base

```
def calculate_tx_better_position_diffraction_based(lpu, htx, wavel, hrx=1.5,
gridp=40):
    # lpu = list position users (nUsers x 2 vector)
    # return: (1x2 vector lat, lon) ex: lpu = [[lat0, lon0], [lat1,lon1]...]
    # passo 1 - calcular caminhos

    lat,lon = apis.extract_lat_lon(lpu)

    minlat = min(lat)
    minlon = min(lon)
    maxlat = max(lat)
    maxlon = max(lon)

    print(f'minlat: {minlat}, minlon: {minlon}, maxlat: {maxlat}, maxlon:
{maxlon}')

    dx = apis.calculate_distance(minlat, minlon, maxlat, minlon)
    dy = apis.calculate_distance(minlat, minlon, minlat, maxlon)
    d1 = sqrt(dx**2 + dy**2)
    d2 = apis.calculate_distance(minlat,minlon,maxlat,maxlon)

    # caso 1: dx > dy
    if dx > dy:
        nptsy = round(sqrt(gridp/round(dx/dy)))
        nptsx = round(dx/dy)*nptsy
        gridp = nptsy * nptsx
    else:
        nptsy = round(sqrt(gridp/round(dy/dx)))
        nptsx = round(dy/dx)*nptsy
        gridp = (nptsy-1) * (nptsx-1)

    debug(gridp)

    xstep = (maxlat-minlat)/(nptsx)
    ystep = (maxlon-minlon)/(nptsy)
    bs_points = [[(minlat + i*xstep,minlon + j*ystep) for j in range(1, nptsy-
1)] for i in range(1, nptsx-1)]
    debug(f'{len(bs_points)} {len(bs_points[0])}')
    debug(f'{nptsx} {nptsy}')

    loss_vector = []
    coordinates_vector = []
    for i in bs_points:
```

```

        for j in i:
            sec_loss_vector = []
            for k in lpu:
                d, h = obtain_terrain_profile([j[0], j[1]], k, 10)
                sec_loss_vector.append(calculate_diffraction_loss(h,d, htx,
wavel, hrx))
            loss_vector.append(sec_loss_vector)
            coordinates_vector.append(j)

    print(f'coordinates_vector: {coordinates_vector}\n\nloss_vector:
{loss_vector}')

    # calcular agora a média em cada vetor:
    meanvec = []
    for l in loss_vector:
        mean = 0
        for i in l:
            mean += i
        meanvec.append(mean)

    print(meanvec)
    print(coordinates_vector[meanvec.index(min(meanvec))])

def calculate_tx_better_position_all_based(lpu, htx, wavel, hrx=1.5,
gridp=40):
    # this model includes diffraction and path loss
    # lpu = list position users (nUsers x 2 vector)
    # return: (1x2 vector lat, lon) ex: lpu = [[lat0, lon0], [lat1,lon1]...]

    lat,lon = apis.extract_lat_lon(lpu)

    # calcular as bordas da matriz:

    minlat = min(lat)
    minlon = min(lon)
    maxlat = max(lat)
    maxlon = max(lon)

    dx = apis.calculate_distance(minlat, minlon, maxlat, minlon)
    dy = apis.calculate_distance(minlat, minlon, minlat, maxlon)
    d1 = sqrt(dx**2 + dy**2)
    d2 = apis.calculate_distance(minlat,minlon,maxlat,maxlon)

    # caso 1: dx > dy

```

```

if dx > dy:
    nptsy = round(sqrt(gridp/round(dx/dy)))
    nptsx = round(dx/dy)*nptsy
    gridp = nptsy * nptsx
else:
    nptsy = round(sqrt(gridp/round(dy/dx)))
    nptsx = round(dy/dx)*nptsy
    gridp = (nptsy-1) * (nptsx-1)

debug(gridp)

xstep = (maxlat-minlat)/(nptsx)
ystep = (maxlon-minlon)/(nptsy)
bs_points = [(minlat + i*xstep,minlon + j*ystep) for j in range(1, nptsy-
1)] for i in range(1, nptsx-1)]
debug(f'{len(bs_points)} {len(bs_points[0])}')
debug(f'{nptsx} {nptsy}')

loss_vector = []
coordinates_vector = []

for i in bs_points:
    for j in i:
        sec_loss_vector = []
        for k in lpu:
            d, h = obtain_terrain_profile([j[0], j[1]], k, 10)
            sec_loss_vector.append(calculate_complete_loss(h,d, htx,
wavel, hrx))
        loss_vector.append(sec_loss_vector)
        coordinates_vector.append(j)

# calcular agora a média em cada vetor:
meanvec = []
for l in loss_vector:
    mean = 0
    for i in l:
        mean += i
    meanvec.append(mean)

return (coordinates_vector[meanvec.index(min(meanvec))])

```