

Um Simulador para Redes Sem Fio sob o Modelo SINR

Fábio Engel de Camargo^{1,2}, Elias P. Duarte Jr.²

¹Universidade Tecnológica Federal do Paraná (UTFPR)
R. Cristo Rei, 19 Toledo 85902-490 PR

²Departamento de Informática – Universidade Federal do Paraná (UFPR)
Caixa Postal 19018 Curitiba 81531-990 PR

fabioe@utfpr.edu.br, elias@inf.ufpr.br

Resumo. *Este trabalho apresenta um simulador para redes sem fio sob o modelo SINR. O modelo SINR considera os efeitos da propagação do sinal (perda de percurso), a existência de ruído de fundo e, principalmente, a interferência entre múltiplos sinais. A ferramenta proposta foi implementada sobre o framework INET e o simulador OMNeT++. São apresentados no artigo cenários simples executados com o objetivo de demonstrar com clareza a validade do simulador, bem como a possibilidade de seu uso para a construção e validação de algoritmos desenvolvidos para este modelo em particular.*

1. Introdução

A modelagem dos efeitos que impactam a comunicação sem fio representa um papel fundamental para o entendimento sobre as comunicações neste tipo de meio e para a criação de protocolos e algoritmos de rede eficientes. A literatura mostra que uma grande quantidade de trabalhos na área de redes sem fio focam em modelos baseados em grafos [Goussevskaia et al. 2010]. Neste tipo de modelo, os dispositivos são representados pelos vértices de um grafo. Se dois vértices (dispositivos) encontram-se sob alcance de transmissão, atribui-se uma aresta indicando a existência de um enlace lógico entre os dispositivos. Entretanto, este modelo não é realmente preciso, pois, não reflete o fato de que os sinais provenientes de dispositivos que encontram-se além de sua vizinhança imediata interferem entre si, impossibilitando inclusive a recepção em alguns casos.

Os modelos baseados em grafo, além de não considerarem os efeitos da superposição dos sinais, tratam o desvanecimento do sinal de maneira binária, como se houvesse uma linha imaginária que delimita o sinal e o interrompe abruptamente a partir de um certo ponto, ignorando o fato de que o sinal desvanece em relação a distância [Halldórsson and Tonoyan 2019]. Devido a estas deficiências, os resultados obtidos por estes modelos frequentemente não são aplicáveis na prática. A utilização de um outro modelo que melhor reflita a realidade, e principalmente, modele adequadamente os efeitos da interferência neste meio, é de fundamental importância para o desenvolvimento de soluções práticas. Neste sentido, o modelo *Signal-to-Interference-plus-Noise Ratio*, ou modelo físico, mostra-se mais adequado para a modelagem do canal de comunicação sem fio. Isso porque considera os efeitos da propagação do sinal (perda de percurso), a existência de ruídos e, principalmente, a interferência entre os múltiplos sinais.

Sob o modelo SINR são propostas diversas estratégias e algoritmos que possuem como alvo a comunicação sem fio. Por exemplo, [Engel and Duarte 2021b,

Saad et al. 2022] apresentam algoritmos de escalonamento construídos sob o modelo SINR. Para esses algoritmos, e muitos outros, é comum a utilização da simulação como mecanismo de avaliação e validação. Dessa maneira, uma ferramenta capaz de simular uma rede de comunicação sem fio sob o modelo SINR é de fundamental interesse. Este trabalho, portanto, descreve um simulador para redes sem fio sob o modelo SINR¹, construído com o OMNeT++² e o arcabouço INET *Framework*³. Apesar de todos os nossos esforços para encontrar outro simulador de redes sem fio sob o modelo SINR disponível, surpreendentemente nenhum foi encontrado. Assim, destacamos a relevância do trabalho, como única opção para simulação deste tipo específico de rede.

O restante do trabalho está organizado da seguinte maneira. A Seção 2 apresenta o modelo SINR. Na Seção 3 é descrita a implementação do simulador SINR. A Seção 4 apresenta resultados que corroboram a validação do simulador. Por fim, as conclusões são apresentadas na Seção 5.

2. O Modelo SINR

Diferentemente de outros modelos que consideram a interferência como um fenômeno binário, o modelo SINR (*Signal-to-Interference-plus-Noise Ratio*), ou modelo físico, captura os efeitos da interferência cumulativa dos transmissores que estão além de sua vizinhança imediata, e introduz a noção de uma “interferência gradativa” [Halldórsson and Tonoyan 2019]. Pesquisas experimentais, como as apresentadas por [Son et al. 2006] e por [Maheshwari et al. 2008] mostram que o modelo SINR não é perfeito, porém, possui boa precisão, principalmente quando comparado a outros modelos.

O modelo SINR estabelece um critério para que uma transmissão seja enviada, recebida e corretamente decodificada pelo receptor. Seja $\ell = (i, j)$ um enlace (*link*), i.e., a representação de uma demanda de comunicação, no qual i é o dispositivo transmissor e j o receptor. No modelo SINR ℓ é factível, ou seja, j recebe corretamente o sinal de i , apenas se satisfizer a inequação exibida pela Equação (1a). Esta inequação determina que a transmissão de i é bem sucedida apenas se a potência do sinal recebido $P_{R(ij)}$ no dispositivo j , em razão da soma das potências dos τ sinais interferentes, mais a potência do ruído de fundo N_0 , for maior ou igual ao limiar γ , definido como SINR *threshold* ou limite SINR. O limite SINR (γ) é expresso em decibéis (dB), e depende de uma série de fatores intrínsecos à tecnologia de comunicação, como por exemplo, o poder de processamento do dispositivo receptor e o esquema de codificação utilizado.

$$\frac{P_{R(ij)}}{N_0 + \sum_{\substack{k=1 \\ k \neq i}}^{\tau} P_{R(kj)}} \geq \gamma \quad (1a)$$

$$\frac{\frac{P_{T_i}}{d(i,j)^\alpha}}{N_0 + \sum_{\substack{k=1 \\ k \neq i}}^{\tau} \frac{P_{T_k}}{d(k,j)^\alpha}} \geq \gamma \quad (1b)$$

A potência do sinal recebida P_R depende do modelo de propagação de ondas de rádio utilizado. Numerosos estudos de propagação do sinal possibilitaram a criação dos modelos de propagação hoje utilizados para prever a cobertura de transmissão em larga

¹Disponível em: <https://github.com/fabioengel/SINR>

²Disponível em: <https://omnetpp.org/>

³Disponível em: <https://inet.omnetpp.org/>

escala para sistemas de comunicação sem fio. A combinação das medições realizadas em campo em função da distância entre transmissor e receptor, junto com a análise teórica possibilitaram a criação de um modelo determinístico geral de propagação para a perda média no percurso [Simon Haykin 2008]. Este modelo indica que a potência do sinal transmitido decai proporcionalmente ao inverso da distância euclidiana entre emissor e receptor, elevados a um expoente denominado de *path loss* (perda de percurso) α . O expoente de *path loss* é uma constante que varia tipicamente entre 2 e 5 e que depende de condições do meio de transmissão, como por exemplo, a existência de obstáculos entre transmissor e receptor [Goussevskaia et al. 2010]. Uma vez que o ganho de potência do canal depende da distância entre os dispositivos no plano, oriunda da geometria euclidiana, este modelo é comumente referenciado como “*geometric SINR model*”, ou modelo SINR geométrico. A inequação que define se $\ell = (i, j)$ é factível no modelo SINR geométrico é apresentada pela Equação (1b). Nesta inequação, utiliza-se P_{T_i} para representar o nível de potência do sinal transmitido por i , e $d(i, j)$ para a distância euclidiana entre os dispositivos i e j . Ainda que o modelo geométrico seja o mais utilizado, o modelo SINR admite a utilização de outros modelos de propagação do sinal, como por exemplo, modelos estocásticos que buscam representar os efeitos do desvanecimento por multipercurso (*multipath fading* ou *multicaminho*).

3. O simulador SINR

O simulador SINR⁴ proposto é uma implementação feita sobre o simulador OMNeT++⁵ e o arcabouço INET Framework⁶. O OMNeT++ (acrônimo para *Objective Modular Network Testbed in C++*) é um simulador de eventos discretos baseado em C++. Seu código é aberto e está liberado para uso sob sem fins lucrativo, sob a licença *Academic Public License*. O OMNeT++ não é propriamente uma plataforma para simulação de redes de computadores. Entretanto, os autores do projeto mantêm um arcabouço de modelos para redes cabeadas e sem fio chamada INET Framework. Destaca-se que não há um módulo específico para o modelo SINR. A seguir, são apresentados os componentes principais da arquitetura do OMNeT++, e conseqüentemente do simulador SINR, bem como a forma que foi implementado utilizando módulos do INET.

3.1. OMNeT++

O desenvolvimento e execução da simulação no OMNeT++ é feito por meio de uma IDE de desenvolvimento (*Integrated Development Environment* ou Ambiente de Desenvolvimento Integrado) baseada na plataforma Eclipse. Esta IDE estende algumas funcionalidades, como por exemplo, a inclusão de novos visualizadores e adiciona funcionalidades para a criação e configuração de modelos (arquivos do tipo NED e INI, por exemplo, descritos a seguir). A IDE permite realizar execuções em lote e apresenta uma ferramenta para análise dos resultados da simulação. Um dos componentes fundamentais para a infraestrutura fornecida consiste na arquitetura baseada em componentes. Nesta abordagem, os modelos são construídos a partir de componentes que podem ser reutilizados, denominados módulos. Assim, os módulos podem ser combinados e também conectados uns aos outros por meio de portas, denominadas *gates*. A comunicação entre os módulos ocorre

⁴Disponível em: <https://github.com/fabioengel/SINR>

⁵Disponível em: <https://omnetpp.org/documentation/>

⁶Disponível em <https://inet.omnetpp.org/docs/developers-guide/index.html>

por meio de troca de mensagens. Para isto, os módulos podem utilizar caminhos pré-definidos por meio dos *gates* e conexões, ou entregar a mensagem diretamente ao módulo destino. Os modelos possuem parâmetros que podem ser utilizados para caracterização e para fins de comportamento, podendo simular ações de dispositivos ou mesmo ambientes como, por exemplo, canais de comunicação. Módulos que encontram-se no nível mais baixo da hierarquia são denominados módulos simples (*simple modules*) e estão programados na linguagem C++.

Além dos módulos simples, compõem o OMNeT++ os descritores de topologia e os definidores de mensagens. Os descritores de topologia são utilizados pelo usuário para descrever a estrutura de um modelo de simulação. Para isto é utilizada uma linguagem própria, denotada como NED (*NETwork Description*). Por meio desta linguagem é possível declarar módulos simples, que podem ser conectados e combinados em módulos compostos. O conteúdo normalmente é salvo dentro de um arquivo com extensão “.ned”. Definidores de mensagens também possuem extensão própria, sendo arquivos salvos com o final “.msg”. Estes definidores permitem a criação de campos de dados para um certo tipo de mensagem. Após a sua criação, estes arquivos são convertidos em classes C++.

A simulação construída pode ser compilada como um programa único e então executada, ou então compartilhada como uma biblioteca. A execução da simulação inicia pela leitura dos arquivos NED, sucedida pelo arquivo de configuração denominado “omnetpp.ini”. Neste arquivo são inseridas configurações de como a simulação deve ser executada, como por exemplo, a definição de valores para alguns parâmetros do modelo. É possível também definir a quantidade de simulações a serem executadas.

A execução da simulação pode ocorrer sob diferentes interfaces para o usuário, conhecidas como ambientes de tempo de execução (*runtime environments*). Atualmente, são suportados os ambientes Cmdenv e Qtenv. O primeiro deles possui uma interface de linha de comando, sendo apropriada para execuções em lote. A segunda é uma interface gráfica baseada na plataforma Qt, onde é possível executar ações passo a passo e ajustar a velocidade da execução. Este ambiente é propício para propósitos de demonstração e atividades educacionais. Os resultados obtidos pelas execuções são salvos em arquivos *result*. Estes arquivos possuem as saídas nos formatos vetoriais e escalares e podem ser lidos pela própria IDE, que possui um ambiente próprio para a análise destes valores.

3.2. INET Framework e Simulador SINR

O *framework* INET⁷ possui suporte para vários modelos de redes de computadores, incluindo redes cabeadas, sem fio, móveis, *ad hoc* e redes de sensores. Possui modelos para a pilha de protocolos da Internet, incluindo implementações, por exemplo, dos protocolos TCP, UDP, IPv4, IPv6, OSPF, BGP, entre outros. Assim como o OMNeT++, são utilizados módulos que comunicam-se por troca de mensagens. Os módulos são combinados de modo a formar dispositivos e componentes de rede como *switches* e roteadores. Sua estrutura permite que novos modelos sejam adaptados para construir novos modelos.

Para a modelagem de um sistema de comunicação sem fio é importante definir um modelo de representação do sinal analógico. O INET trata o sinal analógico como um fenômeno físico que passa pelos processos de: transmissão – o módulo que reside no transmissor cria uma descrição do sinal analógico; propagação – o módulo de

⁷Disponível em: <https://inet.omnetpp.org/Introduction.html>

representação do meio de transmissão (*radio medium module*) aplica os efeitos do desvanecimento sobre o sinal; recepção – o módulo que recebe a representação do sinal, tendo os efeitos da atenuação aplicados, e realiza o cômputo da relação sinal ruído mais interferência no receptor (SINR).

A escolha do modelo de representação do sinal analógico está atrelada aos módulos de transmissão, recepção e meio de transmissão (*radio medium*). O *framework* fornece algumas opções de modelos, a que mais se adapta ao propósito deste trabalho é o modelo *Scalar*. Neste modelo, a potência de transmissão é representada como um valor escalar (possui um número associado a uma unidade de medida). As transmissões possuem frequência central e largura de banda definidos, porém, modelados como sinais planos em frequência e tempo. A atenuação do sinal é calculada, e a SINR utilizada pelo modelo de erros para calcular a probabilidade de recepção do sinal.

Para que o modelo SINR seja simulado algumas alterações sobre o modelo *Scalar* foram feitas. Antes, é preciso informar que o *framework* contém diferentes versões do modelo *Scalar* que adaptam-se a tecnologias distintas, como o Wifi e o Bluetooth. Contudo, optou-se neste trabalho pela utilização de uma versão básica do módulo, denominada *ApskScalarRadioMedium* para representar o meio físico. Trata-se de um módulo que provê um mecanismo de rádio hipotético que simplifica as técnicas de modulação existentes, sem utilizar outras técnicas como as de correção antecipada de erros, transposição e espalhamento. O *ApskScalarRadioMedium* é um módulo composto. Fazem parte de sua composição, entre outros, os módulos: *propagation* - responsável por descrever como o sinal de rádio propaga através do espaço ao longo do tempo; *pathLoss* - determina o nível de redução de potência do sinal à medida que é propagado pelo espaço; *backgroundNoise* - descreve o ruído de fundo e outros ruídos que afetam a qualidade da recepção do sinal.

A modularidade do *framework* permite que os módulos sejam substituídos por outros a fim de adequar um modelo desejado. O módulo de *pathLoss* padrão do *ApskScalarRadioMedium* é referenciado como perda de percurso no espaço livre (*free space path loss*). Este módulo calcula a atenuação da potência do sinal de acordo com a lei do inverso do quadrado ao longo de um único caminho de propagação (*single line-of-sight*). Para isto, considera a utilização de uma antena isotrópica e obtém a perda no espaço livre em função do comprimento de onda irradiado. O *framework* ainda possui outros modelos, como por exemplo, os baseados no desvanecimento Rayleigh e Nakagami, além de muitos outros. Apesar de oferecer muitas opções para a perda de percurso, nenhum dos módulos oferecidos adequa-se para a utilização da construção de uma simulação sob o modelo SINR. Para tal, é necessário a criação de um novo módulo. A este novo módulo é dado o nome de *ModelSinrPathLoss*. Resumidamente, este novo módulo assume que a potência do sinal transmitido decai proporcionalmente ao inverso da distância euclidiana entre emissor e receptor, elevados a um expoente denominado de *path loss* (perda de percurso) α . Deste modo, para $\ell = (i, j)$, a potência recebida em j é igual a $P_{T_i}/d(i, j)^\alpha$.

Ainda é necessário realizar uma outra modificação no modelo *Scalar* para que o critério de correta recepção do sinal seja o mesmo do modelo SINR. Isto porque o módulo de recepção é composto por um módulo denominado *errorModel*. Este módulo realiza o cálculo probabilístico da taxa de erro de bits (*Bit Error Rate* - BER) de cada sinal recebido. O cálculo é feito em função do nível de SINR recebido, quanto menor a relação sinal ruído mais interferência, maior a probabilidade da existência de erro na transmissão.

A utilização deste módulo pode considerar que um sinal com a SINR acima do limite SINR (*threshold*) seja incapaz de ser recebido corretamente, uma vez que o modelo de erros padrão é probabilístico e pode introduzir bits com valores distintos aos transmitidos.

Para que seja construído um ambiente de simulação sob o modelo SINR é necessária a criação de um novo modelo de erros, denominado *ModelSinrError*. Trata-se de um modelo bastante simples que também utiliza como critério a SINR recebida para a simulação de erros no sinal recebido. Porém, este modelo considera que se a SINR do sinal recebido for maior ou igual ao limite SINR, não devem existir erros nos bits recebidos, podendo, portanto, receber corretamente o sinal. Caso contrário, erros são inseridos de acordo com a relação sinal ruído mais interferência e considera-se a recepção impossível de ser feita.

As modificações descritas foram implementadas, e o simulador encontra-se disponível em um repositório público⁸, junto com a documentação, exemplos⁹ e vídeo¹⁰ de apresentação. Desde que foi implementado, o simulador tem sido utilizado em nossa pesquisa sobre algoritmos de escalonamento sob o modelo SINR, com alguns resultados publicados [Engel and Duarte 2021a, Engel and Duarte 2021b]. A seguir, a próxima seção apresenta resultados experimentais que mostram a viabilidade da ferramenta.

4. Resultados Experimentais

Para avaliação do simulador SINR é utilizado um cenário, ilustrado pela Figura 1, que mostra o efeito da interferência cumulativa sobre a recepção do sinal. Neste cenário existem três *links*: (a,b), (c,d) e (f,e). Considera-se o momento em que os dispositivos “a”, “c” e “f” realizam uma transmissão.

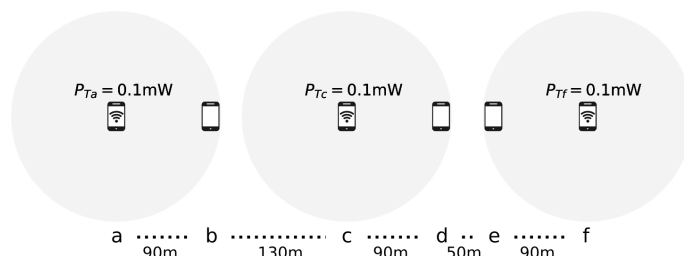


Figura 1. Exemplo de cenário de comunicação sem fio.

Note que ao redor de cada dispositivo transmissor existe uma circunferência que indica o seu alcance de transmissão de 100m. Este alcance é obtido a partir da potência de transmissão, que é uniforme para os três transmissores e igual a 0.1mW; do nível de sensibilidade da antena de recepção, considerado neste exemplo igual a -90dBm; do ruído de fundo igual a -90dB e do *path loss* $\alpha = 4$. Estes valores permitem que a SINR seja calculada para cada respectivo dispositivo receptor do exemplo (“b”, “d” e “e”). Um script¹¹ em Python foi utilizado para este cômputo, o resultado é mostrado pela Tabela 1. Nesta tabela são mostrados para cada *link* o nível de potência recebido P_R do sinal de interesse, a potência de interferência P_I e a SINR resultante do sinal de interesse.

⁸Disponível em: <https://github.com/fabioengel/SINR>

⁹Disponível em: <https://github.com/fabioengel/SINR/tree/master/exemplos/>

¹⁰Disponível em: <https://github.com/fabioengel/SINR/blob/master/arquivos/linkApp.txt>

¹¹Disponível em: <https://github.com/fabioengel/SINR/blob/master/scripts/computoSINR.py>

Tabela 1. Cálculo da SINR para o exemplo da Figura 1.

$\ell = (i, j)$	Potência Recebida de i	Interferência	SINR
(a, b)	$P_{R(a,b)} \cong 1.52\text{pW}$	$P_{I(c,b)} \cong 0.350\text{pW}$ $P_{I(f,b)} \cong 0.00595\text{pW}$ $P_{I(\text{total})} \cong 0.356\text{ pW}$	$\gamma_{ab} \cong 0.50\text{dB}$
(c, d)	$P_{R(c,d)} \cong 1.52\text{pW}$	$P_{I(a,d)} \cong 0.0108\text{pW}$ $P_{I(f,d)} \cong 0.260\text{pW}$ $P_{I(\text{total})} \cong 0.271\text{ pW}$	$\gamma_{cd} \cong 0.78\text{dB}$
(f, e)	$P_{R(f,e)} \cong 1.52\text{pW}$	$P_{I(a,e)} \cong 0.00595\text{pW}$ $P_{I(c,e)} \cong 0.260\text{pW}$ $P_{I(\text{total})} \cong 0.266\text{ pW}$	$\gamma_{fe} \cong 0.80\text{dB}$

A recepção, ou não, destes sinais depende do limite SINR (γ) do dispositivo receptor. Um limite SINR de 1dB, por exemplo, aponta que nenhum dos sinais pode ser recebido corretamente, mesmo o dispositivo receptor estando dentro da área de alcance do sinal de interesse transmitido. Isto porque a interferência cumulativa, oriunda dos outros dois transmissores considerados interferentes a cada *link*, produz um nível de interferência que impede a recepção do sinal de interesse. Este mesmo cenário de exemplo¹² foi reproduzido pelo simulador SINR. Uma imagem do simulador executando em modo gráfico é mostrada pela Figura 2. Na figura é possível verificar, entre outras coisas, os dispositivos dispostos de acordo com o exemplo e o *log* de registro na parte inferior. O registro mostra todas ações realizadas, como, por exemplo, a troca de mensagens entre as camadas e os registros de transmissão e recepção dos sinais com mensagens.

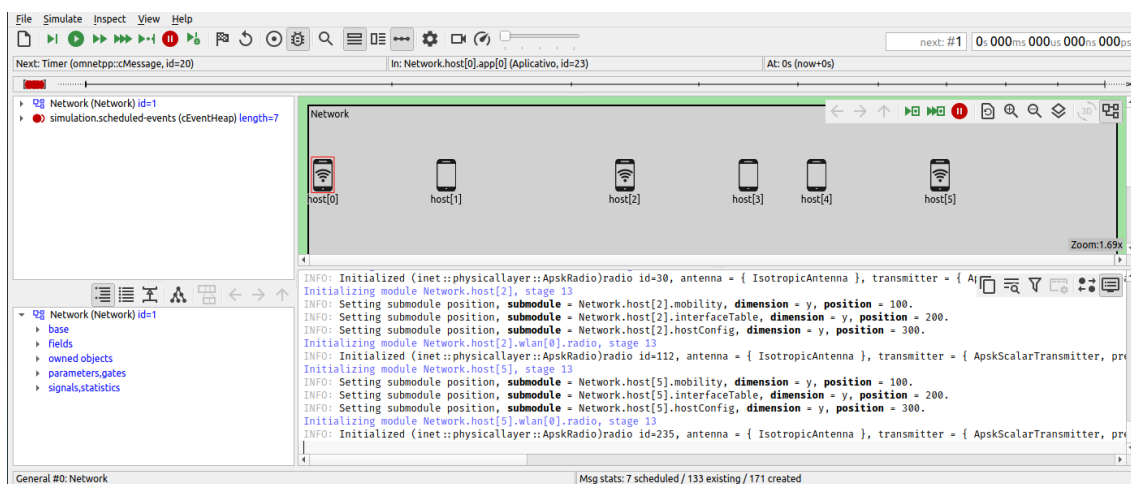


Figura 2. Tela de exemplo do simulador SINR.

Os resultados obtidos pelo simulador SINR correspondem aos valores esperados, i.e., as mensagens são recebidas, ou não, de acordo com o modelo SINR e os níveis de potência de recepção e interferência correspondem aos da Tabela 1. Sobre este exemplo, outros cenários podem ser implementados. Para isto, é disponibilizado um *script*¹³, capaz de gerar arquivos de texto com posições aleatórias no plano para os dispositivos. Este arquivo é lido pela simulação ao iniciar, inserindo os dispositivos em seus respectivos locais. As demais variáveis que compõem o exemplo são especificadas no arquivo de

¹²Disponível em: <https://github.com/fabioengel/SINR/tree/master/exemplos/Exemplo01>

¹³Disponível em: <https://github.com/fabioengel/SINR/blob/master/scripts/geradorCoordenadas.py>

configuração *omnetpp.ini*. Esta simulação possui ainda uma aplicação que permite que sejam especificados quais dispositivos devem transmitir no mesmo instante.

5. Conclusão

O simulador SINR proposto é uma ferramenta que auxilia o desenvolvimento de algoritmos e estratégias de comunicação para redes sem fio sob o modelo SINR. Após buscas exaustivas, não encontramos outro simulador semelhante disponível. A ferramenta consiste em uma implementação realizada com o OMNeT++ e o *framework* INET. Tanto o simulador OMNeT++, quanto o *framework* INET, são ferramentas bem documentadas, com boa difusão e aceitação no meio acadêmico. Isto facilita o entendimento sobre o simulador SINR e conseqüentemente a sua utilização. Sobre o simulador SINR foram apresentados cenários simples, com o objetivo de validar a ferramenta, confirmando que a mesma é capaz de simular o modelo SINR. A modularidade, característica herdada do OMNeT++, permite que alterações sejam realizadas de maneira simples. Por exemplo, modificações futuras podem ser feitas para que além do modelo SINR geométrico, modelos estocásticos sejam utilizados.

Referências

- Engel, F. and Duarte, E. P. (2021a). Argumentos para a inviabilidade prática de uma estratégia de escalonamento para redes sem fio sob o modelo sinr. In *Anais do XXII Workshop de Testes e Tolerância a Falhas*, pages 1–14, Porto Alegre, RS, Brasil. SBC.
- Engel, F. and Duarte, E. P. (2021b). A down-to-earth scheduling strategy for dense sinr wireless networks. In *2021 10th Latin-American Symposium on Dependable Computing (LADC)*, pages 1–6.
- Goussevskaia, O., Pignolet, Y.-A., and Wattenhofer, R. (2010). *Efficiency of Wireless Networks: Approximation Algorithms for the Physical Interference Model*. Now Foundations and Trends.
- Halldórsson, M. M. and Tonoyan, T. (2019). Plain sinr is enough! In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, PODC '19, pages 127–136, New York, NY, USA. Association for Computing Machinery.
- Maheshwari, R., Jain, S., and Das, S. R. (2008). A measurement study of interference modeling and scheduling in low-power wireless networks. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, SenSys '08, pages 141–154.
- Saad, M., El-Moursy, A., Alfawaz, O., Alnajjar, K., and Abdallah, S. (2022). Wireless link scheduling via parallel genetic algorithm. *Concurrency and Computation: Practice and Experience*, 34(6):e6783.
- Simon Haykin, M. M. (2008). *Sistemas Modernos de Comunicações Wireless*. Bookman, Porto Alegre.
- Son, D., Krishnamachari, B., and Heidemann, J. (2006). Experimental study of concurrent transmission in wireless sensor networks. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, SenSys '06, pages 237–250.