

Uma Extensão do ns-3 para Simulação de Transferência Simultânea de Informação e Energia Sem Fio (SWIPT) em Redes IEEE 802.11

José Antônio de França Junior¹, Marcelo M. Carvalho¹

¹Departamento de Engenharia Elétrica
Universidade de Brasília

jose.franca@aluno.unb.br, mmcarvalho@ene.unb.br

Abstract. *This paper introduces an extension to the Network Simulator 3 (ns-3) to enable Simultaneous Wireless Information and Power Transfer (SWIPT) operation over IEEE 802.11 networks. We provide a detailed description about the proposed extension regarding its major building blocks, its underlying mechanisms, and its response to physical layer (PHY) parameters and modes of operation. Further, we discuss the development of new classes and the implementation of new Application Programming Interfaces (API) that were implemented on some already existing member functions. The outcome of this investigation is a full-featured SWIPT simulation environment that can be used for the design and performance evaluation of new SWIPT-based energy-efficient protocols based on IEEE 802.11 technology.*

Resumo. *Este trabalho apresenta uma extensão para o simulador ns-3 que permite a simulação da tecnologia de transferência simultânea de informação e energia sem fio (SWIPT) em redes IEEE 802.11. Para tanto, são apresentados os elementos constitutivos da proposta de extensão, seus mecanismos e sua resposta a diferentes parâmetros e modos de operação da camada física. Além disso, são apresentados os detalhes da implementação de novas classes, como também as alterações realizadas em funções-membro de classes já existentes com o objetivo de acomodar novas APIs. O ambiente de simulação proposto permite o desenvolvimento e a avaliação de desempenho de novos protocolos baseados na integração da tecnologia SWIPT às redes IEEE 802.11, visando sua operação energeticamente eficiente.*

1. Introdução

No atual cenário socio-econômico de elevado grau de dependência tecnológica, em que se evidencia a contraposição de forças antagônicas representadas pela crise energética global e pelas crescentes projeções sobre a quantidade de dispositivos móveis e de Internet das Coisas (IoT, do Inglês: *Internet of Things*), de acordo com [Bedi et al. 2018], urge adotar fontes de energia renováveis, tais como o SWIPT, de modo a assegurar a independência energética dos atuais e futuros dispositivos.

Assim, o conceito do SWIPT pode ser compreendido como uma junção de tecnologias pertencentes a diferentes áreas do conhecimento que por razões diversas têm sido abordadas de forma isolada, a saber, a transferência de informação em redes sem fio (WIT, do Inglês: *Wireless Information Transfer*) e a transmissão de energia elétrica sem fio, ou,

por meio do espectro eletromagnético (WET, do Inglês: *Wireless Energy Transfer*), mas no contexto da denominação original do SWIPT referem-se à execução simultânea do WIT e WET sobre um mesmo sinal eletromagnético recebido pela antena do receptor.

Com base na revisão da literatura científica atual, não foi possível identificar a existência de artigos que contemplem estudos sobre o desenvolvimento de modelos computacionais voltados para a simulação do SWIPT em redes IEEE 802.11. Portanto, o presente estudo constitui contribuição inovadora, com extenso campo de aplicação, como por exemplo, validação de modelos analíticos, auxílio no desenvolvimento de novos projetos de pesquisa e outros.

Além disso, uma característica ímpar apresentada pela presente proposta de extensão do ns-3 refere-se à possibilidade de sua utilização em quaisquer tecnologias de rede do padrão IEEE 802.11, quais sejam, IEEE 802.11a/b/g/n/ah, em virtude de sua estrutura modular e de seu desenvolvimento com foco na redução de dependências entre classes, que serão detalhados adiante. De forma prática, a seleção de diferentes padrões IEEE 802.11 pode ser realizada por meio da função `SetStandard` da classe `WifiHelper` que é independente do módulo de energia utilizado para simular o consumo de energia dos nós.

As seções subsequentes deste artigo apresentam informações organizadas de acordo com a seguinte estrutura: Na Seção 2 são apresentados conceitos básicos de SWIPT. A Seção 3 contempla descrição detalhada da implementação do SWIPT no simulador de redes ns-3. Na Seção 4 apresentam-se e discutem-se os resultados. Finalmente, na Seção 5 apresentamos a conclusão e trabalhos futuros.

2. Conceitos Básicos de SWIPT

Em uma perspectiva ampla, a arquitetura de um receptor SWIPT pode ser definida a partir de seus aspectos macroestruturais e microestruturais. No âmbito dos aspectos macroestruturais, a arquitetura de um receptor SWIPT é formada por dois blocos principais, a saber, o decodificador de informações e o coletor de energia. A inclusão do decodificador de informações nesta definição macroestrutural decorre da necessidade de implementação de elementos adicionais para manutenção do sincronismo entre os blocos e possibilitar a recepção de informações paramétricas provenientes do coletor de energia.

Já no que se refere aos aspectos microestruturais, a arquitetura de um receptor SWIPT baseado na divisão de potência é composta por: antena receptora, utilizada na recepção de sinais de radiofrequências (RF); fator de divisão de potência (θ_s), utilizado para fracionar a potência do sinal de RF entre os blocos do decodificador de informações e do coletor de energia; circuito casador de impedância, utilizado para compatibilizar o valor da impedância dos elementos; retificador de corrente alternada (AC, do Inglês: *Alternating Current*) para corrente contínua (DC, do Inglês: *Direct Current*), cuja nomenclatura dispensa definições e é representado por (η), e conversor DC-DC, utilizado para majorar a eficiência da transferência energética para a recarga da bateria (β). Assim, de acordo com [Luo et al. 2021], o nível de tensão do sinal de RF aplicado à entrada do decodificador de informação, após a incidência de θ_s , pode ser modelado por

$$y_s = \sqrt{\theta_s} \left(\frac{\sqrt{P} h x_s}{\sqrt{1 + d^\alpha}} + n_a \right) + n_p, \quad (1)$$

em que h corresponde ao ganho do canal, x_s representa o símbolo transmitido com potência unitária, d é a distância entre o transmissor e o receptor, e n_p , é o ruído de processamento. Em decorrência disso, o nível de energia coletado pelo SWIPT pode ser modelado por

$$E_s = \eta(1 - \theta_s) \left(\frac{P|h|^2}{1 + d^\alpha} + \sigma_a^2 \right) T_s, \quad (2)$$

em que T_s corresponde à duração de um símbolo e σ_a^2 corresponde ao ruído da antena. Finalmente, o nível de energia destinado à recarga da bateria pode ser modelado por [Luo et al. 2021]

$$E_b = \beta E_s = \beta \eta (1 - \theta_s) \left(\frac{P|h|^2}{1 + d^\alpha} + \sigma_a^2 \right) T_s, \quad (3)$$

a qual refere-se à energia coletada a partir de um único símbolo recebido. Naturalmente, este cálculo deve ser estendido à recepção de um pacote inteiro de dados, como descrito na seção a seguir.

3. Implementação do SWIPT no ns-3

A implementação do modelo do SWIPT envolveu a realização de atividades de planejamento, desenvolvimento em linguagem C++ e análise de resultados na plataforma ns-3, que resultaram no desenvolvimento das seguintes classes: `SwiptHarvester`, `SwiptPhyListener`, e na função de ajuda `SwiptHarvesterHelper`.

Adicionalmente, fez-se necessária a inclusão de novas APIs ou parâmetros nas seguintes classes já existentes no módulo `./src/wifi/model` da árvore de subdiretórios de códigos fonte: `YansWiFiPhy`, `YansWiFiChannel`, `InterferenceHelper`, `WiFiPhy` e `WifiPhyStateHelper`, como também no módulo `./src/energy/model`, no código fonte da classe `LiIonEnergySource`, com o objetivo de permitir o intercâmbio de informações paramétricas entre objetos da PHY e as novas classes.

Assim, esclarece-se que a classe `SwiptHarvester` representa o bloco do coletor de energia e a classe `YansWiFiPhy` o decodificador de informações. Já o `SwiptHarvesterHelper` refere-se à função de ajuda que agrega parâmetros de configuração da classe `SwiptHarvester`. A classe `YansWiFiChannel` refere-se a uma abstração do canal de RF; `InterferenceHelper` é uma função de ajuda aos mecanismos internos da classe `YansWiFiPhy`; `WifiPhyStateHelper`, apesar de sua nomenclatura, possui declarações em seu código fonte que a classificam no rol de classes do ns-3; a classe `WiFiPhy` também é uma abstração da PHY, cujo arquivo de cabeçalho define os estados de máquina; e a classe `LiIonEnergySource` representa o modelo de uma bateria constituída por íons de lítio (Li-Ion, do Inglês, *Lithium-Ion*).

Ademais, em razão da necessidade de estabelecimento de sincronismo para permitir a operação simultânea do SWIPT nos moldes de sua denominação, faz-se necessário que, entre as classes `SwiptHarvester` e `YansWiFiPhy`, seja realizada a troca de informações sobre os estados em que a máquina de estados da PHY se encontra em determinado momento, quais sejam, recepção (RX), transmissão (TX), economia de energia (SLEEP), verificação de disponibilidade do canal de RF (CCA_BUSY), estado de espera (IDLE), troca de canal (SWITCHING) e desligado (OFF). A Fig. 1(a) apresenta

uma perspectiva da interconexão entre estas classes com a PHY. Nas subsecções a seguir os detalhes de cada função serão apresentados.

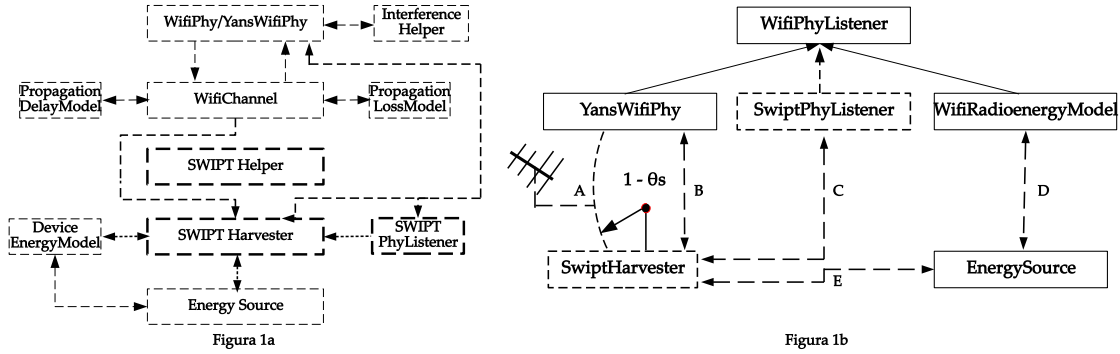


Figura 1. Diagrama de blocos representativo da interconexão entre PHY e SWIPT.

3.1. Construção de um Esqueleto e Adequação dos Parâmetros

Conforme mencionado na introdução deste artigo, a extensão do ns-3 proposta pode ser utilizada em quaisquer redes do padrão IEEE 802.11 disponíveis na versão 3.23 do ns-3. Esta característica modular ocorre em virtude do baixo nível de dependências entre classes, que por sua vez é proveniente do emprego da metodologia de *Scaffolding*, de acordo com o manual do ns-3 versão 3.23 [ns-3 project 2017]. O emprego desta metodologia resultou no desenvolvimento da classe denominada *SwiptHarvester*. Os arquivos de código fonte e de cabeçalho desta nova classe foram incluídos no subdiretório `./src/energy` da árvore `./src`.

Uma característica relevante sobre o funcionamento do ns-3 observada durante o desenvolvimento deste projeto refere-se à unidade de dados de protocolo (PDU, do Inglês: *Protocol Data Unit*) da PHY, que de acordo com o trabalho de [Lacage and Henderson 2006] é representado por quadros, em razão do ns-3 não contemplar a transmissão e recepção de sinais eletromagnéticos para fins de simulação de redes. Consequentemente, o parâmetro referente à duração de símbolos não pertence ao escopo do ns-3, resultando na necessidade de adequação das Equações (2) e (3), de acordo com as expressões a seguir, respectivamente,

$$E_s = \eta(1 - \theta_s)P_k T_k, \quad (4)$$

e

$$E_b = \beta\eta(1 - \theta_s)P_k T_k, \quad (5)$$

em que P_k refere-se à potência do quadro na saída da antena e T_k a duração dos quadros.

3.2. PHY Listener e Máquina de Estados

Para promover o sincronismo entre as tarefas de WIT e WET, implementou-se a subclasse *SwiptPhyListener* dentro do código fonte da classe *SwiptHarvester*, utilizando-se o mecanismo de herança de atributos e métodos da classe *WifiPhyListener*, conforme apresentado nos diagramas das Figs. 1(a) e 1(b). Assim, a classe *WifiPhyListener* possui a função de *Listener* sobre os diferentes

contextos da máquina de estados da PHY. Dessa forma, quando a classe `YansWifiPhy` realiza a transição para um novo estado, a informação sobre este estado é difundida a todos os *Listeners* compreendidos nos limites de um mesmo nó.

Ressalta-se que a operação mencionada no parágrafo anterior possui escopo limitado ao interior de cada nó da rede. Portanto, em condições normais de operação, não há transferência de informações sobre os estados da PHY entre dispositivos. Além disso, para que um objeto possa se tornar elegível à execução de atividades de *Listener*, o endereço de memória que armazena dados de sua instanciação deve ser registrado junto à classe `WifiPhy` por meio de ponteiros. Desta forma, quando a classe `YansWifiPhy` realizar a difusão de notificações de estado a todos os *Listeners*, apenas aqueles que se encontrarem registrados receberão as notificações.

Ademais, esta proposta contemplou a inclusão do estado OFF no cabeçalho da classe `YansWifiPhy`, como também dos métodos necessários para transição da máquina de estados da PHY na função de ajuda `WifiPhyStateHelper`, de modo que, em caso de eventual esgotamento da energia da bateria, o nó possa divulgar sua condição de desligado e o coletor de energia possa promover a recarga.

3.3. Coletor de Energia *SwiptHarvester*

Conforme mencionado no início desta seção, a classe `SwiptHarvester` representa o coletor de energia SWIPT. Para tanto, contém atributos, métodos e outros, relativos ao funcionamento deste objeto. Sua representação em diagrama de blocos, em conjunto com outras dependências é apresentada na Fig. 1(b).

Note-se que neste diagrama, as classes referentes à proposta de extensão deste artigo são representadas por blocos com linhas tracejadas, enquanto a antena e a chave conectada à sua saída são apenas parâmetros utilizados nestas classes. Assim, na Fig. 1b, do lado esquerdo do diagrama, o elemento que simboliza a antena está conectado a uma chave que representa o parâmetro θ_s , cujo objetivo é fracionar a potência do quadro recebido em duas partes e, após isso, encaminhá-las para os respectivos blocos do decodificador de informações e do coletor de energia, representados pelas classes `YansWifiPhy` e `SwiptHarvester`, respectivamente. Contudo, em razão do objetivo desta representação estar voltada para o `SwiptHarvester`, a expressão localizada acima da chave representa o complemento de θ_s , de acordo com as Equações (4) e (5).

Na Fig. 1(b), a conexão representada pela linha *A*, em formato semicircular, que conecta o `SwiptHarvester` ao `YansWifiChannel`, representa a ligação entre o canal de RF e o coletor de energia. Para tanto, foi implementada uma API para realizar a passagem dos parâmetros relativos aos quadros recebidos para o `SwiptHarvester`, no mesmo instante de tempo em que estas informações também são passadas para a classe `YansWifiPhy`.

Já a conexão simbolizada pela linha *B* representa a API utilizada pela classe `YansWifiChannel` para obter o valor de θ_s , que constitui um dos atributos configuráveis da classe `SwiptHarvester`, repassado à classe `YansWifiPhy` para fins de cálculo do valor de potência do quadro a ser recebido pelo decodificador. A conexão representada pela linha *C* alude às notificações encaminhadas sobre a máquina de estados da PHY e informações para fins de registro do `SwiptPhyListener`.

A conexão representada pela linha E refere-se à API por meio da qual a classe `SwiptHarvester` realiza a transferência de energia coletada para a classe `LiIonEnergySource`. Destaca-se que esta transferência ou recarga só pode ocorrer mediante o atendimento de duas condições concomitantes, quais sejam, o recebimento da solicitação de transferência de energia enviada pela classe `LiIonEnergySource` e somente durante o intervalo de tempo em que a máquina de estados da PHY estiver no estado RX. Por fim, os blocos apresentados na parte superior do diagrama da Fig. 1(b), quais sejam, `SwiptPhyListener`, `YansWifiPhy` e `WifiRadioEnergyModel` representam classes herdeiras dos atributos da classe pública `SwiptPhyListener`, conforme explicado na subseção anterior.

4. Resultados

Apresentam-se nesta seção os resultados obtidos por meio de simulações realizadas em uma rede com arquitetura ponto-a-ponto, constituída por um AP e um nó. Assim, utilizou-se a implementação que contempla o padrão IEEE 802.11ah desenvolvida por [Tian et al. 2018], sobre a qual a extensão proposta neste trabalho foi implementada.

4.1. Configuração do Cenário de Testes

Neste cenário apenas o nó foi configurado para realizar a coleta energia utilizando o SWIPT. Assim, instalou-se um servidor UDP em sua camada de aplicação para permitir a recepção de tráfego saturado, cujo *payload* dos pacotes UDP foi de 1000 *bytes* a uma taxa de transmissão de 150 kbps. No que se refere ao AP, em sua camada de aplicação foi configurado o respectivo *UDP Client*, responsável pela geração do tráfego saturado.

Os principais parâmetros de configuração do sistema são apresentados na Tabela 1. Dentre os quais, destacam-se θ_s e a distância entre o AP e o nó. Em relação ao parâmetro θ_s , seus valores são apresentados na Tabela 1. Esta faixa é subdividida em dois intervalos sucessivos, sendo o primeiro de 0 a 0,1, em que o valor imediatamente superior a 0 é 10^{-10} . Após isso, os valores subsequentes são obtidos por meio da multiplicação sucessiva dos valores imediatamente anteriores por 10. O segundo intervalo varia de 0,2 a 1,0, com passos de 0,1. No que se refere à faixa de valores utilizados para a distância, esta possui passo incremental de 1 metro, dentro dos limite inicial e final de 1,5m e 10,5m, respectivamente. Portanto, para cada combinação entre os valores de θ_s e distância, a simulação foi executada por 300 segundos.

4.2. Relação Sinal-Ruído

O gráfico da Fig. 2(a), à esquerda, apresenta as curvas da relação sinal-ruído (SNR, do Inglês: *Signal to Noise Ratio*) em resposta aos valores de θ_s e da distância, conforme mencionados na subseção anterior. Assim, verifica-se que, mesmo com uma redução do nível da potência do quadro recebido em 90 dB, quando o valor de θ_s é igual à 10^{-9} , representado pela curva de menor nível na parte inferior do gráfico, em quase toda a sua extensão ao longo do eixo das abcissas, os resultados dos valores de SNR são suficientes para garantir que o nó realize recepção de informação quase sem perdas.

As perdas de pacotes ocasionadas pela degradação de SNR ocorrem apenas nas situações em que o valor de θ_s é igual à 10^{-9} e a distância é superior a 8,5m. O mesmo

Tabela 1. Parâmetros de Configurados nas Simulações do SWIPT no ns-3

Parâmetro no ns-3	Descrição	Valor
dist	Distância entre o nó e o AP	1,5 a 10,5 m
psFactor	Fator de Divisão de Potência	0,0 a 1,0
apTxPower	Potência Transmissão do AP	30 dBm
txPower	Potência Transmissão do Nó	0 dBm
apTxGain	Ganho da Antena do AP	6 dBi
rxAntennaGain	Ganho da Antena do Nó	6 dBi
sourceInitialEnergy	Energia Inicial na Bateria	5 J
InitialCellVoltage	Tensão Inicial da Célula de Li-ion	1 v
BeaconInterval	Intervalo de Beacon	102.400 μ s
TxCurrentA	Corrente DC TX	8,5 mA
RxCurrentA	Corrente DC RX	6 mA
SleepCurrentA	Corrente DC SLEEP	1,2 μ A
IdleCurrentA	Corrente DC IDLE	6 mA
CcaBusyCurrentA	Corrente DC CCA_BUSY	6 mA
SwitchingCurrentA	Corrente DC SWITCHING	8,5 mA
payloadSize	Carga Útil de Pacotes UDP	1024Bytes
DataRate	Taxa de TX de Quadros de Informação	150 Kbps @ 1MHz
BasicRate	Taxa de TX de Quadros de Controle	300 Kbps @ 1MHz
Seed	Semente Num. Pseudoaleatórios	1

efeito é também verificado na vazão da rede, que apresenta taxa máxima de bits transmitidos por segundo em quase toda a extensão dos valores de θ_s e distância, exceto para os casos mencionados.

4.3. Energia Total Coletada e Energia Total Consumida

O gráfico da Fig. 2(b) apresenta curvas de energia coletada em oposição ao nível de energia consumida pelo nó, considerando-se os mesmos valores de θ_s e da distância, conforme delineado em subseção anterior. Assim, verifica-se que as curvas possuem decaimento similar ao caso anterior. Contudo, neste gráfico, a linha vermelha tracejada na direção horizontal representa o consumo de energia do nó para quaisquer combinações de θ_s e distância, que permanece constante em todo o intervalo.

Portanto, a partir destas informações, verifica-se que os valores de energia coletada (representados pelos pontos das curvas acima da linha do consumo de energia) correspondem à região em que ocorre a sustentabilidade energética. Já para os valores inferiores à linha vermelha, não há sustentabilidade energética.

5. Conclusão e Trabalhos Futuros

Este artigo apresentou uma extensão do ns-3 para viabilizar a simulação do coletor de energia SWIPT em redes IEEE 802.11. Assim, foram apresentados conceitos, detalhes da implementação e resultados dos experimentos realizados, que foram posteriormente validados por meio de cálculos teóricos. Portanto, conclui-se que a extensão proposta neste artigo por meio do coletor de energia SWIPT apresenta atributos factíveis para avaliação da sustentabilidade energética de redes IEEE 802.11.

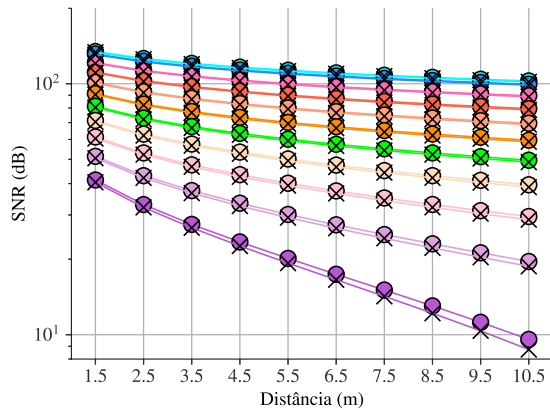


Figura 2a: Curvas de SNR para valores de θ_s . Valores obtidos por meio de simulação (Sim.) e cálculos teóricos (Calc.)

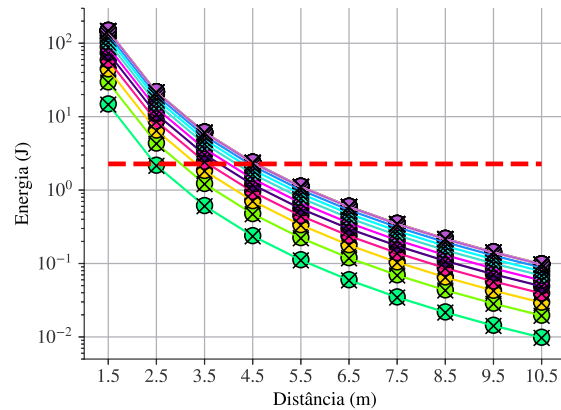


Figura 2b: Curvas de energia coletada para valores de θ_s , obtidas por simulação e cálculos teóricos. Linha tracejada horizontal, em vermelho, representa o consumo de energia do sensor

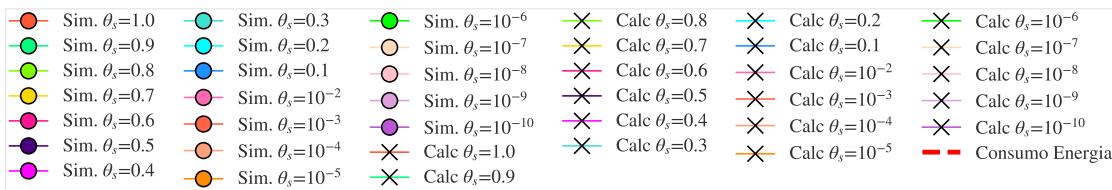


Figura 2. SNR e Energia Coletada vs. Consumo

Em trabalhos futuros serão apresentados resultados a partir da avaliação do coletor de energia SWIPT em cenários de arquiteturas de redes com múltiplos nós para permitir a análise dos possíveis efeitos que possam ser ocasionados pela contenção no acesso ao meio, colisões de quadros e outros. Ainda, tem-se por objetivo realizar o estudo de algoritmos que possam maximizar a sustentabilidade energética por meio da alocação dinâmica de números de identificação das estações para formação de grupos distintos em redes IEEE 802.11ah.

6. URL de Acesso à Ferramenta, ao Manual e ao Vídeo

Todo o programa desenvolvido pode ser acessado em [OneDrive](#).

Referências

- Bedi, G., Venayagamoorthy, G. K., Singh, R., Brooks, R. R., and Wang, K.-C. (2018). Review of Internet of Things (IoT) in electric power and energy systems. *IEEE Internet of Things Journal*, 5(2):847–870.
- Lacage, M. and Henderson, T. R. (2006). Yet another network simulator. In *Proceedings from the 2006 workshop on ns-2: the IP network simulator*, pages 12–es.
- Luo, Y., Luo, C., Min, G., Parr, G., and McClean, S. (2021). On the study of sustainability and outage of swipt-enabled wireless communications. *IEEE Journal of Selected Topics in Signal Processing*, 15(5):1159–1168.
- ns-3 project (2017). *ns-3 Manual Release ns-3.23*. [Acesso em: 17/03/2023].
- Tian, L., Sljivo, A., Santi, S., De Poorter, E., Hoebeke, J., and Famaey, J. (2018). Extension of the IEEE 802.11ah ns-3 simulation module. In *Workshop on ns-3 (WNS3)*.