

MiniManager: Uma Proposta de Plataforma Web para Emulação de Redes Heterogêneas de Comunicação Móvel

André M. Demori¹, Renan Rodrigues Moreira Resende da Silva¹,
Pedro Paulo Leal de Albuquerque¹ David Fernandes Cruz Moura¹,
Júlio César Cardoso Tesolin¹, Maria Cláudia Reis Cavalcanti¹

¹Instituto Militar de Engenharia (IME)
22290-270 – Rio de Janeiro – RJ – Brasil

{andredemori, renan.silva, pedro.albuquerque, david, jcctesolin, yoko}@ime.eb.br

Abstract. *During the development of communication networks, experimentation and data analysis assists researchers and decision-makers in defining the most appropriate transmission, reception, and routing strategies. One of the methods used to reproduce the reality of communication networks is the network emulation, providing inputs for reaction and performance analysis of the system under test. However, some gaps were identified while emulating a network process. This article presents the MiniManager, an application based on Mininet-WiFi that facilitates the management and the configuration of experiments through an intuitive interface and a system capable of capturing the origin of the data resulting from the emulations.*

Resumo. *Durante o desenvolvimento de redes de comunicação, a experimentação e a análise de dados auxiliam os pesquisadores e os tomadores de decisão na definição das melhores estratégias de transmissão, recepção e roteamento. Um dos métodos utilizados para reproduzir a realidade das redes de comunicação é a emulação, provendo insumos para avaliar a reação e o desempenho do sistema sob teste. Diante de algumas lacunas identificadas no processo de emulação de redes, este artigo apresenta o MiniManager, uma aplicação baseada no Mininet-WiFi que facilita a gestão e a configuração de experimentos, por meio de uma interface intuitiva e um sistema capaz de capturar a proveniência dos dados resultantes das emulações.*

1. Introdução

Para enfrentar os desafios encontrados na construção de redes de comunicação em cenários heterogêneos, é necessário entender os conceitos e paradigmas adentrados nesse domínio de aplicação. As redes heterogêneas têm como principal desafio a busca por interoperabilidade, dada a diversidade de sistemas, equipamentos, tecnologias e participantes envolvidos no processo de comunicação. A evolução das redes de comunicação móveis nas últimas quatro décadas revolucionou a forma como a humanidade interage. Como resultado, o projeto e a operação de tais redes tornou-se dependente de soluções mais complexas baseadas na flexibilidade e na capacidade cognitiva de seus nós. Propostas tecnológicas como as Redes Definidas por Software (SDN) [Kreutz et al. 2015] e os Rádios Definidos por Software (RDS) são parte deste esforço para enfrentar tal complexidade [Kreutz et al. 2015].

O RDS também ganha relevância no contexto das redes de comunicações móveis militares ou redes táticas, no sentido de que não se faz uso de um ponto central de interoperabilidade e, portanto, provê soluções mais seguras em operações de combate. Em 2012, o Ministério da Defesa criou o programa RDS-Defesa [Prado Filho et al. 2017] para fomentar o desenvolvimento de soluções de RDS. Além do desenvolvimento do dispositivo, também são necessários experimentos para avaliar o comportamento em situações onde as Forças Armadas podem operar.

A experimentação de redes de comunicação móvel é um assunto recorrente no ambiente acadêmico. Existem alternativas de experimentação que vão dos testes em ambientes de ensaio, passando por emuladores e simuladores. Por exemplo, as Redes Definidas por Software (SDN), que têm como principal característica a separação do plano de controle do plano de dados, têm sido cada vez mais utilizadas no enfrentamento de problemas voltados para redes táticas. O cenário de experimentação de redes de comunicações móveis em ambientes militares também é explorado por diversos pesquisadores, usando plataformas de experimentação conhecidas, como o simulador ns-3 [Mazzini et al. 2010] e emulador Mininet-WiFi [Zhao et al. 2019]. Contudo, algumas lacunas foram identificadas no período de avaliação de tais ferramentas, como a falta do registro dos experimentos por meio de processos de captura da proveniência dos dados dos experimentos e da falta de um ambiente amigável de configuração e visualização do experimento para o pesquisador.

Assim, o MiniManager é apresentado neste trabalho como uma proposta de solução de gerenciamento de experimentos de sistemas de comunicação móvel, baseado nas funcionalidades do emulador Mininet-WiFi [Fontes et al. 2015] [Fontes and Rothenberg 2016] [dos Reis Fontes and Rothenberg 2019] e construído sobre uma plataforma Web, propiciando aos pesquisadores maior agilidade e praticidade no processo de experimentação e na análise dos dados gerados.

Este artigo está estruturado da seguinte forma: a Seção 2 apresenta conceitos básicos e trabalhos relacionados, enquanto a Seção 3 descreve a arquitetura da solução e o modelo conceitual de dados. A Seção 4 descreve a implementação do MiniManager e, finalmente, a Seção 5 apresenta as conclusões e os trabalhos futuros.

2. Conceitos Básicos e Trabalhos Relacionados

2.1. Simuladores, Emuladores e Ambientes de Ensaio

Como descrito por Ramon Fontes [Fontes 2018], a pesquisa na área de redes de comunicação móvel possui algumas alternativas para avaliar e analisar o comportamento dos nós e dos enlaces, assim como validar protocolos de rede. Dentre os principais métodos estão a simulação, a emulação e a realização de testes em ambientes de ensaio (*testbeds*).

Os simuladores representam uma forma de avaliar o comportamento dos sistemas de comunicação, oferecendo ao pesquisador uma grande flexibilidade aliada a um alto grau de controle e repetibilidade, com baixos custos se comparado a outras soluções. Contudo, não são capazes de modelar com profundidade todos os detalhes da rede. Isso deriva da dificuldade de conseguir fidelidade suficiente em tarefas de mobilidade do nó, uma vez que a simulação se baseia em um sistema hipotético que precisa ser modelado,

gerando resultados que podem não ser realistas. Além disso, o código usado em uma simulação tende a não ser compatível com aqueles que poderiam ser executados em redes de comunicação móvel reais.

Um ambiente de ensaio consiste em um grupo de dispositivos reais usados para testar e avaliar um sistema de comunicação, provendo acesso compartilhado com hardware dedicado aos pesquisadores, isolando fisicamente tais dispositivos de forma a prevenir que os diversos experimentos afetem uns aos outros. Em um ambiente de ensaio, as dúvidas sobre a veracidade das medições são reduzidas. Contudo, a grande desvantagem do ambiente de ensaio é que o hardware real é necessário e tem que ser mantido. Além de problemas como restrição de acesso e controle limitado sobre o ambiente, os experimentos são de difícil repetição e as configurações experimentais são restritas tanto em tamanho quanto em complexidade.

Como contraponto, os emuladores combinam tanto características de simuladores quanto de implementações reais, utilizando a implementação em teste em um contexto parcialmente sintético, tornando possível aumentar a veracidade do ambiente de estudo, oferecendo uma maneira flexível de mimetizar as propriedades de diversos tipos de redes. Assim, a emulação representa um esforço para resolver as deficiências da simulação por meio da interação com o mundo real, mantendo seus pontos fortes como a repetibilidade e a facilidade de configuração [Fontes 2018].

Uma extensa pesquisa foi realizada por Ramon Fontes em sua tese de doutorado [Fontes 2018] para identificar as alternativas mais atuais de experimentação de redes de comunicação sem fio. Como exemplos, é possível citar o OM-Net++ [Varga and Hornig 2008] e o OpenNet [Chan et al. 2014] como representantes dos simuladores, WARP ¹ e o EMULAB ² como representantes de ambientes de ensaio e o ns-3 [Mancini et al. 2014] e o Mininet-WiFi [Fontes et al. 2015] [Fontes and Rothenberg 2016] [dos Reis Fontes and Rothenberg 2019] como representantes dos simuladores e emuladores respectivamente. Segundo o autor, o resultado da comparação das diversas alternativas de experimentação de redes de comunicação sugere a utilização do Mininet-WiFi como uma solução adequada para a realização de tais atividades. O emulador Mininet-WiFi, é dedicado a emulações de redes sem fio, seguindo o paradigma das Redes Definidas por Software com divisão entre plano de dados e plano de controle, sendo uma extensão do emulador Mininet [Lantz et al. 2010]. Não obstante, durante o processo de construção de redes para experimentação no Mininet Wi-Fi, pode haver a necessidade de registrar os resultados e as configurações das topologias para análise e reprodução. Além disso, o processo de configuração das redes pode se tornar trabalhoso, uma vez que para construções de redes rebuscadas, o usuário terá que escrever códigos na linguagem Python.

2.2. Proveniência

A emulação de sistemas de comunicações móveis nada mais é do que a construção de um ambiente capaz de avaliar uma determinada hipótese. A proveniência dos dados torna-se pertinente nesse contexto, pois consiste na informação sobre entidades, atividades e pessoas envolvidas na geração de um conjunto de dados [Moreau and Missier 2013],

¹<http://warpproject.org/trac>

²<https://www.emulab.net/portal/frontpage.php>

que podem comprovar uma hipótese. No caso do processo de emulação, ao capturar a proveniência, registram-se não só o resultado final, mas também as etapas, passos e tentativas, como as variações de parâmetros em cada rodada durante todo o processo de experimentação. Contudo, um único experimento pode não ser capaz de entregar os dados necessários para avaliar uma hipótese colocada. Assim, é comum a execução de múltiplos experimentos para fortalecer a avaliação, representando diferentes cenários que possuem, normalmente, diferentes atributos e/ou condições iniciais.

Segundo Sérgio Cruz [Cruz 2011], a proveniência de dados abrange a representação, o registro e a recuperação precisa das informações sobre as atividades feitas por um ou mais agentes em um determinado contexto. Em [Freire et al. 2008], os autores classificam a proveniência em tarefas computacionais de duas formas: retrospectiva e prospectiva. Enquanto a primeira aborda a coleta dos dados gerados por um experimento executado, a segunda trata da sequência de etapas realizadas para a captura de um dado, bem como as condições iniciais. Dessa forma, o MiniManager, em seu estado atual, foca na captura da proveniência retrospectiva durante a execução dos experimentos.

Há várias abordagens para a construção de um modelo que represente a proveniência, como a PROV-DM [Moreau and Missier 2013]. Ela define uma estrutura base a ser estendida de forma a atender as especificidades de cada requisito, consistindo de três elementos básicos: Entidades, Atividades e Agentes. As Entidades são representações de conceitos ou objetos com atributos fixos, que são manipuladas por operações realizadas ao longo de um período de tempo, também chamadas de Atividades. Assim, as Atividades podem se relacionar com as Entidades tanto para a utilização desses recursos (leitura das informações, para comunicação com outras Atividades por exemplo) quanto para a sua criação. Por fim, os Agentes são elementos que possuem a responsabilidade pela execução das Atividades. Consequentemente, pode-se registrar a geração de dados em um processo, como sendo a manipulação de Entidades (criação, atualização e atribuição) em Atividades atreladas aos seus Agentes.

3. Arquitetura do MiniManager

O desenvolvimento de qualquer sistema computacional pressupõe o levantamento de requisitos, sejam eles funcionais ou não funcionais. Após entrevistas com alguns pesquisadores usuários do Mininet-Wifi, foram identificados alguns conceitos, como **plano de teste**, **versão** e **rodada**, além de outras necessidades, mormente aquelas relacionadas à guarda dos experimentos e da interação dos pesquisadores com o programa.

O **plano de teste** representa uma configuração de rede, que pode ser infraestruturada ou *ad hoc*. Cada plano de teste possui pelo menos uma versão. Em cada **versão**, é necessário configurar as características dos elementos de rede (potência de transmissão, *bitrate*, perda por propagação, entre outros), o comportamento dos elementos de rede (movimentação das estações móveis, rompimentos de enlaces, perda de energia da estação, entre outros) e os testes de desempenho que as estações móveis devem realizar (latência, taxa de transmissão, entre outros). A **rodada** representa uma simulação feita pelo pesquisador, seguindo as configurações definidas pela versão e plano de teste. A Figura 1 apresenta parte modelo conceitual produzido para a solução, representando a relação entre conceitos próprios do Mininet-WiFi, encapsulados na entidade *Configuração*, e os conceitos próprios da emulação como o plano de teste e versão.

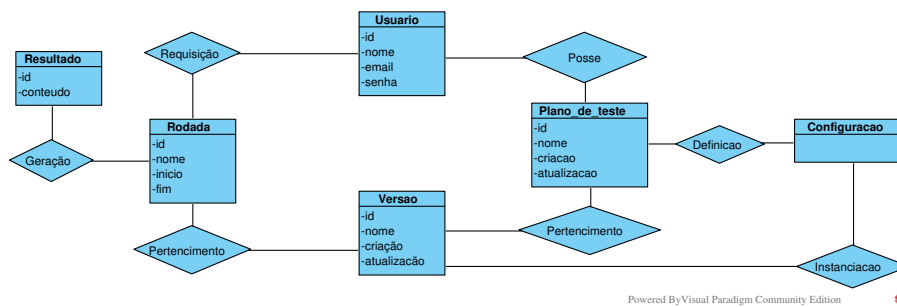


Figura 1. Modelo Conceitual Simplificado - MiniManager

Os casos de uso do MiniManager podem ser divididos em dois blocos: o primeiro, trata da gestão dos experimentos pelo usuário, como a manutenção dos planos de teste e suas versões. Já o segundo bloco abrange os casos de uso responsáveis pela realização de uma emulação e pela manutenção das informações de cada rodada de um determinado plano de teste. Desta maneira, o MiniManager foi construído como um conjunto de componentes, como mostra a Figura 2. O pesquisador interage com a plataforma por meio de dois módulos: *Configurator* e *Experimenter*. O módulo *Configurator* recebe os dados de configuração da rede na interface do usuário para criação do **plano de teste** e de suas **versões**. Já o módulo *Experimenter* gerencia os experimentos, onde a emulação de rede é executada e o recebimento de requisições de novas rodadas de experimento são realizadas por meio de uma interface com o usuário.

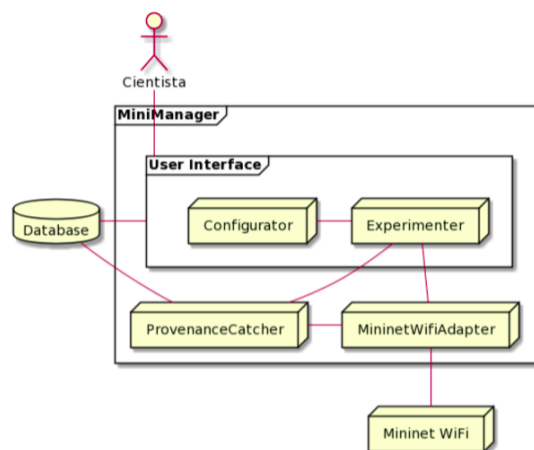


Figura 2. Componentes da Arquitetura

O módulo *Experimenter* comunica-se com o módulo *ProvenanceCatcher*, que é responsável pela coleta dos dados gerados em cada rodada, além de permitir a recuperação dos dados de experimentos já executados. O módulo responsável pela comunicação com o Mininet Wi-Fi é o *MininetWifiAdapter*, onde os dados de configurações da rede são passados como entrada e o envio dos resultados gerados pelo Mininet-WiFi é realizado em tempo real para os outros módulos do sistema.

4. Implementação e Execução

A plataforma foi implementada seguindo o padrão MVC (Model-View-Controller), utilizando Django Framework, que utiliza a linguagem Python. Foi utilizado como sis-

tema gerenciador de banco de dados o PostgreSQL, sendo responsável pelo repositório das configurações e experimentos realizados. O emulador Mininet-WiFi é integrado à aplicação para executar as emulações de rede com as configurações passadas pelo usuário pela interface. O processo de desenvolvimento da aplicação foi realizado pensando em uma fácil manutenção dos requisitos funcionais do sistema por parte dos desenvolvedores, assim como prover interface mais amigável para os usuários.

A interface web para configuração das redes foi desenvolvida em formato de formulário. Portanto, o primeiro passo na aplicação é criar um plano de teste, onde são definidos o nome do plano, o autor e uma breve descrição. Logo em seguida, são listados os planos de teste já criados e o usuário deve escolher aquele no qual serão criadas versões para a execução dos experimentos. A página de configuração de versão é responsável pela definição dos nós da rede, dos links entre os nós, dos modelos de propagação e mobilidade, além de estabelecer quais parâmetros de medida de rádio frequência e desempenho serão coletados. Alternativamente, as configurações também podem ser carregadas por meio de um arquivo JSON, uma vez que a aplicação permite exportar e importar as configurações para facilitar a reprodutibilidade dos experimentos.

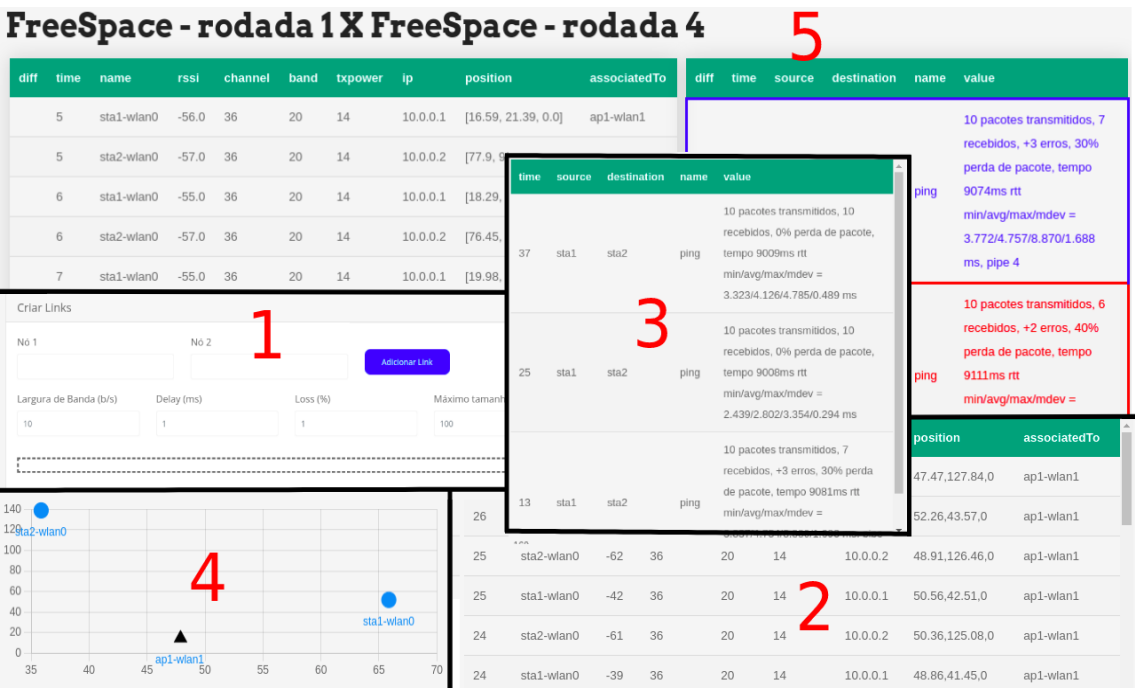


Figura 3. Telas do MiniManager

Após a criação da versão, o usuário é direcionado a uma página com todas as versões daquele plano de teste. Posteriormente, o usuário deve abrir a versão desejada e clicar em executar para iniciar o experimento. A Figura 3 mostra quatro diferentes telas acessadas pelo usuário. A Tela indicada com o número 1 mostra um trecho da tela de criação do plano de teste, onde o usuário tem acesso a um formulário para preencher os dados referentes às configurações da rede, como dos links entre os nós, como indicado na imagem. Já a Tela 2 mostra, em tempo real, as informações de rádio-frequência dos nós, enquanto a Tela 3 apresenta estatísticas relacionadas à entrega de pacotes. A Tela 4 apresenta em tempo real uma representação da mobilidade dos nós durante a execução de

um experimento. Finalmente, a tela número 5 mostra a etapa de comparação entre duas rodadas que foram executadas.

A aplicação está disponível para a comunidade em uma instância Amazon EC2. O código do projeto é aberto e pode ser encontrado no seguinte link: <https://gitlab.com/andredemori/minimanager>. No repositório pode-se encontrar um arquivo de configuração com o nome *Free Space.json* para configurar uma rede com dois nós e um ponto de acesso.

5. Conclusão

Os sistemas de emulação de redes exigem uma forte demanda por soluções que disponibilizem ao pesquisador um alto grau de abstração dos componentes e do ambiente a ser emulado. Emuladores como o Mininet Wi-Fi permitem ao usuário criar ambientes de rede baseado no paradigma das Redes Definidas por Software, possibilitando que o usuário configure cada componente da rede.

Não obstante, o processo de coleta de resultados e de configuração da topologia pode se tornar uma tarefa trabalhosa quando se tem um alto número de variações de parâmetros e de dados a serem coletados. Durante o processo de planejamento e de construção de redes, podem ser realizadas exaustivas experimentações para se chegar a uma quantidade de dados adequada para se construir as melhores estratégias de configuração, sendo que a falta de gerenciamento nesse tipo de cenário pode conduzir a falhas e erros no processo de experimentação. O MiniManager, aplicação apresentada neste trabalho, tem como objetivo ser uma plataforma web que ajuda na construção de redes móveis através do Mininet Wi-Fi, com uma interface intuitiva para o usuário e com um sistema de proveniência de dados dos experimentos que facilitam no momento de comparar resultados entre as rodadas de experimentação da rede.

Através do MiniManager, os usuários podem ter na web um local para construir emulações de redes móveis com o poder de abstração do Mininet Wi-Fi e de forma prática, sem a necessidade de escrita de códigos. Além disso a exportação das configurações também permite a fácil reprodutibilidade dos planos de teste configurados. Por conseguinte, essa ferramenta tem por objetivo contribuir com a comunidade de redes no sentido de trazer mais praticidade e facilidade no gerenciamento e configuração de experimentos de redes sem fio.

Agradecimentos

Agradecemos o apoio dado pelo convênio FINEP/DCT/FAPEB (ref.: 2904/20 convênio 01.20.0272.00) para a realização desse trabalho.

Referências

- Chan, M.-C., Chen, C., Huang, J.-X., Kuo, T., Yen, L.-H., and Tseng, C.-C. (2014). Opennet: A simulator for software-defined wireless local area network. pages 3332–3336.
- Cruz, S. (2011). *An Approach to Support The Management of Data Provenance of Scientific Experiments*. PhD thesis, Universidade Federal do Rio de Janeiro, Rio de Janeiro-RJ-Brasil.

- dos Reis Fontes, R. and Rothenberg, C. E. (2019). Mininet-wifi: Plataforma de emulação para redes sem fio definidas por software. In *Anais Estendidos do XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 201–208. SBC.
- Fontes, R. D. R. (2018). *Mininet-Wifi: Emulation Platform For Software-Defined Wireless Networks*. PhD thesis, Universidade Estadual de Campinas, Campinas-SP-Brasil.
- Fontes, R. d. R. and Rothenberg, C. E. (2016). Mininet-wifi: A platform for hybrid physical-virtual software-defined wireless networking research. In *Proceedings of the 2016 ACM SIGCOMM Conference*, pages 607–608.
- Fontes, R. R., Afzal, S., Brito, S. H., Santos, M. A., and Rothenberg, C. E. (2015). Mininet-wifi: Emulating software-defined wireless networks. In *2015 11th International Conference on Network and Service Management (CNSM)*, pages 384–389. IEEE.
- Freire, J., Koop, D., Santos, E., and Silva, C. T. (2008). Provenance for computational tasks: A survey. *Computing in Science & Engineering*, 10(3):11–21.
- Kreutz, D., Ramos, F. M. V., Veríssimo, P. E., Rothenberg, C. E., Azodolmolky, S., and Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76.
- Lantz, B., Heller, B., and McKeown, N. (2010). A network in a laptop: rapid prototyping for software-defined networks. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, pages 1–6.
- Mancini, E. P., Soni, H., Turlatti, T., Dabbous, W., and Tazaki, H. (2014). Demo abstract: realistic evaluation of kernel protocols and software defined wireless networks with dce/ns-3. In *MSWiM '14*.
- Mazzini, A., Stefanelli, C., Tortonesi, M., Benincasa, G., and Suri, N. (2010). Disservice: Network state monitoring and prediction for opportunistic information dissemination in tactical networks. In *2010-MILCOM 2010 MILITARY COMMUNICATIONS CONFERENCE*, pages 555–560. IEEE.
- Moreau, L. and Missier, P. (2013). Prov-dm: The prov data model. *W3C Recommendation*.
- Prado Filho, H. V., Galdino, J. F., and Moura, D. F. C. (2017). Pesquisa e desenvolvimento de produtos de defesa: reflexões e fatos sobre o projeto rádio definido por software do ministério da defesa à luz do modelo de inovação em tríplice hélice. *Rio de Janeiro: RMCT*, 34(1):6–19.
- Varga, A. and Hornig, R. (2008). An overview of the omnet++ simulation environment. In *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems Workshops*, Simutools '08, Brussels, BEL. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
- Zhao, Q., Brown, A. J., Kim, J. H., and Gerla, M. (2019). An integrated software-defined battlefield network testbed for tactical scenario emulation. In *MILCOM 2019-2019 IEEE Military Communications Conference (MILCOM)*, pages 373–378. IEEE.