

# Projeto de Validação do Simulador MANNASim

Igor C. Uchôa , Thais R. M. Braga Silva (Orientadora) , Fabrício A. Silva  
(Co-orientador)

<sup>1</sup>Instituto de Ciências Exatas e Tecnológicas  
Universidade Federal de Viçosa (UFV-Campus Florestal)  
Florestal, MG - Brasil

**Abstract.** *Wireless Sensor Networks (WSNs) have been explored to be used in the Internet of Things (IoT). There are three basic strategies to evaluate WSN solutions: analytical analysis, simulation and real experimentation. The first one requires less cost, but is less precise. The last one, on the other hand, represents the real network, but requires a huge amount of investment. Therefore, simulation is the most adopted strategy in the literature. Among the existing simulators, MANNASim is one of the most adopted. However, MANNASim has not been validated so far. Thus, the objective of this work is to validate the MANNASim simulator by comparing its results with real ones. With the conclusion of this work, it is expected that MANNASim can continue to be a constant target of researchers.*

**Resumo.** *As Redes de Sensores Sem Fio (RSSFs) vêm sendo exploradas para serem parte da Internet das Coisas (IoT). Para que as soluções propostas para esse tipo de rede sejam avaliadas, pesquisadores podem adotar três estratégias: análise analítica, simulação e experimentos reais. A primeira é menos custosa, mas apresenta um nível de precisão mais baixo. A última, por outro lado, representa a rede real, mas requer um alto investimento financeiro. Sendo assim, a simulação é a estratégia mais utilizada. Dentre os simuladores existentes, o MANNASim vem sendo bastante utilizado. Porém, até o momento, esse simulador ainda não foi validado. Portanto, o objetivo deste trabalho é validar o simulador MANNASim, comparando os resultados de saída do mesmo com resultados de experimentos reais. Com a efetivação desse resultado, espera-se que o MANNASim possa ser ainda mais utilizado pela comunidade acadêmica.*

## 1. Introdução

As ciências tecnológicas, dentre elas a computação, são indispensáveis para o cotidiano do ser humano, desde a realização de tarefas simples até as mais complexas. Para tal, há uma infinidade de vertentes originárias do termo computação, que têm como objetivo a atuação direta na vida das pessoas simplificando o máximo de esforço possível, seja físico ou mental e as ajudando na tomada de decisões. Como exemplo, pode-se citar a inteligência artificial (IA), o aprendizado de máquina, o processamento de dados e as redes de computadores, como sendo de grande importância na atualidade. Essas subvertentes são reunidas e condensadas nas poderosas Redes de Sensores sem Fio (RSSFs) [Akyildiz et al. 2002].

Uma RSSF, diferentemente das redes tradicionais, é uma rede formada por um grande número de nodos (sensores) que possuem restrições de energia e devem ser otimizadas para se auto configurarem e se adaptarem a problemas na falha de comunicação e perda de nodos [Loureiro et al. 2003]. É possível citar uma gama de aplicações para as RSSFs, tais como o sensoreamento da umidade de florestas densas, de difícil acesso; o monitoramento de formas de vida em vulcões; a avaliação da saúde de um paciente através da coleta de componentes sanguíneos dentre inúmeras outras situações.

Nos últimos anos, as RSSFs têm aumentado a sua popularidade dentre a comunidade Científica da Tecnologia da Informação e têm levado muitos pesquisadores a investir tempo e recursos para que haja um incremento contínuo de conhecimento acerca dessa área. Grande parte desses avanços são direcionados para experimentos que visam explorar o poder dos nós sensores na realização de atividades que, para um ser humano, seriam trabalhosas ou até mesmo impossíveis, como visto no parágrafo anterior. Além disso, as RSSFs são uma parte importante da chamada Internet das Coisas (IoT), que visa conectar os mais variados dispositivos.

Um dos principais desafios encontrados por pesquisadores na atualidade está na avaliação de novas soluções para RSSFs, que devem optar por um modelo real, analítico ou uma simulação de seu modelo previamente estabelecido. As duas primeiras citadas apresentam algumas características indesejadas para quem deseja coletar resultados acerca da precisão de sua solução: (1) o custo para se ter uma rede real é alto; (2) *testbeds*<sup>1</sup> existentes são restritos em escala e configuração; (3) a avaliação analítica é pobre em detalhes. Portanto, a técnica mais utilizada para avaliações de soluções para RSSFs é a simulação via softwares [Araújo et al. 2009].

Visando a simulação em diferentes cenários - com configurações e características distintas - de um experimento para RSSF, tem-se o *MANNASim* [Braga et al. 2004], um simulador implementado sobre a camada de aplicação, com as camadas inferiores estabelecidas pelo *Network Simulator ou NS-2* [UC Berkeley et al. 1995]. O NS-2 é um software amplamente utilizado para testar e simular cenários diversos em uma rede de computadores convencional que, através de scripts, torna possível descrever as configurações dos experimentos. Com a implementação do *MANNASim*, o NS-2 passou a dar suporte às RSSFs permitindo a pesquisadores desse ramo utilizar suas bases sólidas em conjunto com o mesmo para atingirem suas metas.

Durante o período em que o *MANNASim* estava sendo desenvolvido, ainda não havia meios para que fosse validado. Não haviam recursos suficientes para montar uma rede física e não existiam *testbeds* para testar o experimento. Como trata-se de um trabalho específico que lida com uma área crítica, o *MANNASim* deve ser validado. Assim, pode-se definir como o objetivo desse trabalho, o teste e validação dessa nova extensão, através da comparação dos resultados de um mesmo experimento para o simulador e para uma rede física. Para se alcançar esse objetivo, um mesmo cenário será aplicado em uma RSSF real e no *MANNASim*, e os resultados serão comparados. É esperado que os resultados sejam similares, comprovando assim a confiabilidade do simulador.

Este texto está organizado da seguinte forma. Na seção 2 serão mostrados alguns trabalhos relacionados ou similares ao *MANNASim* e alguns dentre os vários *testbeds*

---

<sup>1</sup> Ambiente físico para execução remota de experimentos acadêmicos

existentes que mais se aproximam do utilizado durante o trabalho. Na seção 3 serão abordados alguns tópicos acerca das funcionalidades e diferenciais do MANNASim em relação aos outros simuladores. Na seção 4, será descrito toda a metodologia utilizada na validação do MANNASim, desde os detalhes do cenário até as métricas utilizadas. Na seção 5, os resultados serão mostrados, bem como as suas análises posteriores. Por fim, na seção 6, serão apresentadas as considerações finais e trabalhos futuros.

## 2. Trabalhos relacionados

Com o surgimento do MANNASim, é possível destacar outros trabalhos que realizam simulações similares em RSSFs e que foram validados, embora utilizem abordagens diferentes.

- **TOSSIM**: em [Levis and Lee 2003], é um simulador de RSSFs disponível na literatura que recebe como entrada o firmware do nó sensor, ou seja, o código TinyOS diretamente. Esse simulador dá suporte a imperfeições, tempo global e implementa algumas camadas do modelo OSI. Com o TOSSIM, porém, há restrições quanto aos tipos de nós suportados e seus respectivos sistemas operacionais, dificultando um pouco o trabalho de usuários de outras plataformas
- **Cooja**: em [Sehgal 2013], é um simulador de RSSFs elaborado para suportar vários tipos de nós, com diferentes sistemas operacionais. Além disso conta com uma interface gráfica (GUI) para acompanhar em tempo real, os resultados das simulações. O Cooja recebe como entrada um arquivo executável das plataforma TinyOS ou Contiki e o atribui aos nós definidos pelo usuário. É uma ferramenta extensível, porém, ainda limitada no que diz respeito a suporte de novas plataformas.
- **MSPSim**: em [Eriksson et al. 2008], é um emulador específico para o nó MSP430 e é capaz de simular algumas plataformas específicas. Implementa as interfaces para os LEDs, o Beeper, o rádio e o serial. Simula o comportamento de um nó apenas. Ideal para testar funções que envolvam envio de dados a porta serial.
- **OMNet++**: em [Varga 1993], ou *Objective Modular Network Test-bed in C++*, é um simulador de redes escrito em C++ e montado através de uma linguagem de alto nível para descrição de topologia (NED - *Network Description*). Possui uma interface gráfica (GUI) e integração com a ferramenta Eclipse IDE [Eclipse Foundation]. Para simular uma RSSF, há extensões que utilizam as bases do OMNet++ para operar, como o Castalia [Pediaditakis et al. 2010], um simulador genérico que estima o comportamento dos nodos de modo geral, com alguns elementos de uma rede sem fio convencional, como o protocolo MAC 802.11. Não dá suporte para nós específicos com características distintas.
- **Avrora**: em [Titzer et al. 2005] é um simulador de RSSFs implementado em Java com suporte a executáveis de diferentes sistemas operacionais. É capaz de simular os microcontroladores AVR para os nós Micaz e Atmel. Implementa o paralelismo a nível de multithreading, podendo aumentar o desempenho para grandes simulações. Como os trabalhos citados acima, é específico para essas duas plataformas, apenas.
- **ATEMU**: em [Polley et al. 2004], ou *Atmel Emulator*, é um simulador de RSSFs implementado em C, com suporte para os nós Micaz. Possui uma interface gráfica para depuração do código e seu funcionamento interno, bem como seu desempenho, são similares ao TOSSIM.

Neste trabalho, os resultados de um simulador são comparados com os resultados de uma rede real representada em um testbed. Portanto, além dos simuladores, também é importante descrever alguns dos testbeds existentes:

- **MoteLab:** em [Werner-Allen et al. 2005] é um testbed público via interface web com 190 nós programáveis em NesC, com sistema operacional TinyOS. Permite ao usuário interagir com cada um dos nós sensores individualmente e submeter códigos fonte compilados para serem executados nos mesmos. Também apresenta uma interface gráfica para acompanhamento das execuções agendadas.
- **ORBIT:** ou *Open Access Research Testbed for Next-Generation Wireless Networks* em [Raychaudhuri et al. 2005] é um testbed que permite ao usuário emular protocolos e aplicações em tempo real, podendo especificar a topologia da rede e o tipo de conexão entre os nós. Possui 400 nós com tecnologia IEEE 802.11.
- **Indryia:** em [Doddavenkatappa et al. 2012], é um ambiente remoto para alocação de experimentos em RSSF localizado na escola de computação da *Universidade Nacional de Singapura - NUS*. Nele, têm-se a disposição, 139 nós TelosB equipados com sensores infravermelho, de luminosidade, de temperatura e de umidade, distribuídos pelos três andares da universidade (ver Figura 1). Através do Indryia Testbed, é possível submeter o executável a partir do código em nesC, compilado para o sistema operacional TinyOS. É permitido também escolher quais nós irão executar a aplicação e por quanto tempo (limitado de 5 a 30 minutos). Ao término de cada execução, o testbed fornece um log de saída com os dados enviados pelos nós sensores para a porta serial.

### 3. O Simulador MANNASim

O MANNASim, é um simulador de Redes de Sensores Sem Fio detalhado em [Manna Research Group et al. 2002] e continuamente aprimorado por pesquisadores de RSSFs da atualidade. É um framework escrito em C++ e TCL, criado para estender as funcionalidades do NS-2, dando suporte para estabelecer e simular uma rede de sensores, levando em consideração todas as suas limitações e tecnologias disponíveis.

A ferramenta suporte para o MANNASim, o *Network Simulator*, é uma ferramenta de código aberto escrita em C++ e para simular uma rede, utiliza scripts em OTcl para criação e descrição dos cenários. O NS-2 implementa as camadas inferiores do modelo OSI, deixando a camada de aplicação modular e extensível, permitindo a outros pesquisadores a criação de extensões para incrementar cada vez mais o software, utilizando os seus alicerces, firmemente estabelecidos.

Para realizar o seu propósito, o MANNASim [Braga et al. 2004] estende algumas classes de entidade do NS-2 que representam os nós da rede e algumas classes de controle que fazem o gerenciamento da energia, do processamento e dos dados trocados entre eles. O NS-2 também fornece interfaces para o tempo de simulação, acesso aos protocolos de roteamento internos e seus respectivos logs de saída, o que torna o MANNASim modular, com implementações apenas na camada de aplicação.

Além disso, de acordo com [Braga et al. 2004], é possível destacar alguns requisitos implementados exclusivamente no MANNASim e que são desejáveis para um simulador de RSSF, tais como: (1) A simulação de diferentes tipos de sensores, tais como tem-

peratura, umidade, luminosidade e outros; (2) a possibilidade de utilizar diferentes opções para o sensoriamento e a disseminação dos dados pela rede - contínua, programada ou sob demanda - de forma independente; (3) a simulação da rede com uma determinada hierarquia, permitindo vários níveis hierárquicos; (4) a simulação de vários pontos de acesso; (5) a possibilidade de se implementar uma rede heterogênea, composta por diferentes nós; (6) a provisão de uma base para diferentes tipos de processamento dos dados pelos nós sensores e (7) a utilização de diferentes protocolos para as camadas mais inferiores da pilha, não afetando o funcionamento do MANNASim.

A ferramenta geradora de scripts (*Script Generation Tool*) do cenário e suas características é o front end do MANNASim, pelo qual são definidas as diretrizes do experimento, tais como número de nós, aplicação, protocolos de roteamento, características dos sensores, suas coordenadas, topografia e outros. Esse arquivo, por sua vez é interpretado pelo simulador, que inicia a simulação com os parâmetros definidos e retorna, ao final, um log de saída, com dados úteis, tais como o consumo de energia, número de mensagens e pacotes enviados, recebidos ou perdidos.

#### 4. Metodologia de validação

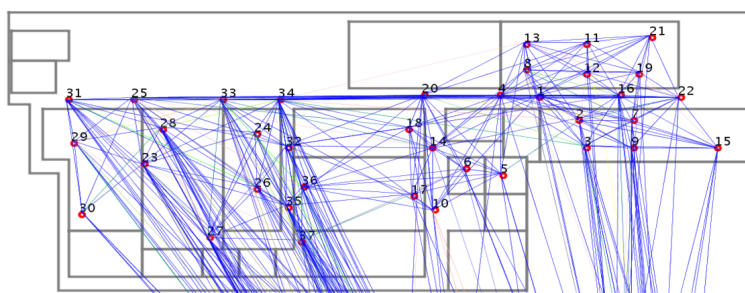
A validação do simulador MANNASim será realizada através da execução de um mesmo experimento para dois ambientes distintos, sendo um deles o MANNASim e o outro um Testbed específico de RSSFs. Após as duas execuções, ambos os resultados serão comparados e assim, teremos uma ideia de o quão próximos estão as duas plataformas.

O Testbed escolhido para validar o MANNASim foi o *Indryia Testbed* [Doddavenkatappa et al. 2012]. Alguns detalhes do Indryia Testbed chamaram a atenção em relação aos demais e foram fundamentais para sua escolha para o trabalho, tais como, (1) a facilidade de uso, (2) a praticidade e usabilidade na realização dos trabalhos, (3) a necessidade de se implementar um arquivo MIG - *Message Interface Generator* - podendo-se escolher os dados a serem enviados para a porta serial e (4) a disponibilidade dos servidores nas submissões dos trabalhos, podendo colher os resultados alguns minutos após a execução dos experimentos.

Para montar o experimento no testbed, foi escolhido o cenário contido no piso 1 da escola de computação da *Universidade Nacional de Singapura - NUS* (ver Figura 2), contendo os nós sensores de identificadores 1, 3 (ponto de acesso), 5, 6, 8 e 9. Esses são alguns dos nós que se localizam próximos ao raio de alcance do nó ponto de acesso, minimizando a necessidade de uso de um protocolo de roteamento. Um executável do sistema operacional TinyOS simulando o comportamento de um nó sensor de temperatura foi atribuído a todos os 6 nós em uma janela de tempo de 5 minutos.



Figura 1. Distribuição de alguns nós sensores no Indryia Testbed [Doddavenkatappa et al. 2012]



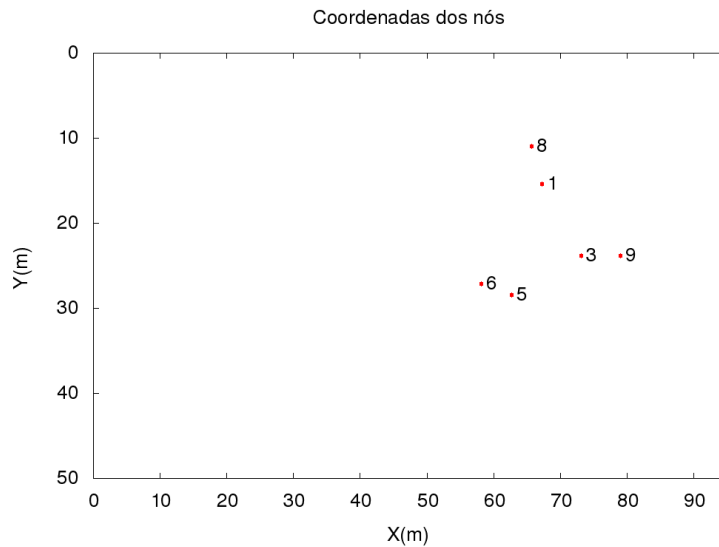
**Figura 2. Nós sensores do piso 1 do Indryia Testbed [Doddavenkatappa et al. 2012]**

O cenário no MANNASim foi construído através da ferramenta *Script Generation Tool*, responsável por gerar o script em TCL, com as configurações do experimento de acordo com as características dos nós sensores utilizados no testbed. Os detalhes podem ser vistos na tabela 1.

**Tabela 1. Descrição do cenário no MANNASim**

Tamanho do cenário	90mx50m
Quantidade de nós sensores convencionais	5
Quantidade de nós ponto de acesso	1
Sensing Power	0.015W
Processing Power	0.024W
Instruções por segundo	8MIPS
Consumo com TX	0.036W
Consumo com RX	0.024W
Taxa de transmissão	28.8Kbps
Tipo de sensoriamento	Programada
Intervalo para sensoriamento	1s
Tipo de disseminação	Programada
Intervalo para disseminação	1s
Protocolo de roteamento	DSDV
Procolo da camada MAC	IEEE 802.15.4
Tempo de simulação	5min (300s)

Algumas modificações no script TCL gerado foram feitas, como a aproximação das coordenadas dos nós em relação ao testbed para que houvesse o máximo de fidelidade possível. Para tal, verificou-se que o tamanho aproximado do menor retângulo que contém todos os nós do piso 1 era de aproximadamente 90m x 50m. Assim uma proporção foi feita para chegar aos valores aproximados de suas localizações.



**Figura 3. Reprodução do piso 1 da escola de computação da Universidade Nacional de Singapura no MANNASim**

Com os experimentos moldados para cada plataforma, foram definidas algumas métricas consideradas importantes na comparação das execuções e validação do MANNASim. São elas:

- **Atraso médio da rede:** é uma métrica que revela quanto tempo, em segundos, as mensagens levaram da saída do nó transmissor, até a chegada ao nó receptor;
- **Consumo de energia:** é uma métrica que revela qual o consumo de energia, em Volts, durante a transmissão, recepção e processamento de mensagens;
- **Mensagens enviadas:** é uma métrica que revela quantas mensagens foram enviadas pelos nós sensores com destino ao ponto de acesso.

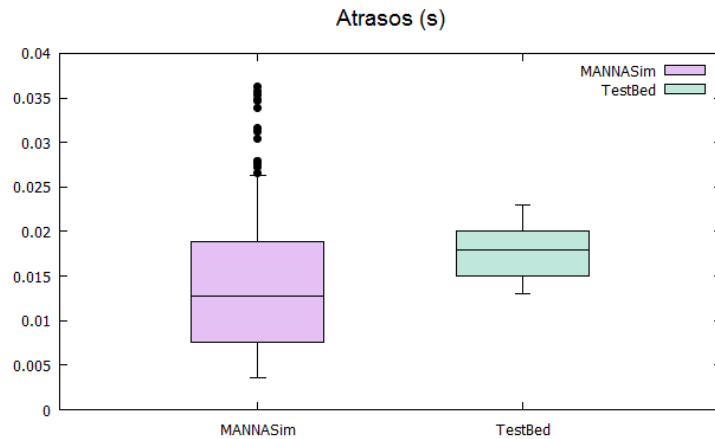
Embora haja outras métricas para se avaliar, foram selecionadas essas que são consideradas críticas para uma RSSF e que um bom simulador deve implementá-las de modo realístico para dar ao pesquisador uma visão geral de seu experimento.

## 5. Resultados

Implementadas as metodologias definidas na seção anterior, são mostrados agora os resultados das comparações para as métricas escolhidas por meio de estruturas estatísticas denominadas *boxplots*<sup>2</sup>. Para isso, foram realizados 5 execuções do experimento no *testbed* e uma simulação no MANNASim.

- **Atraso médio da rede:**

<sup>2</sup>Boxplots são gráficos capazes de avaliar a efetiva distribuição dos dados, sendo formados pelo 1º e 3º quartis, mediana, limite superior e limite inferior

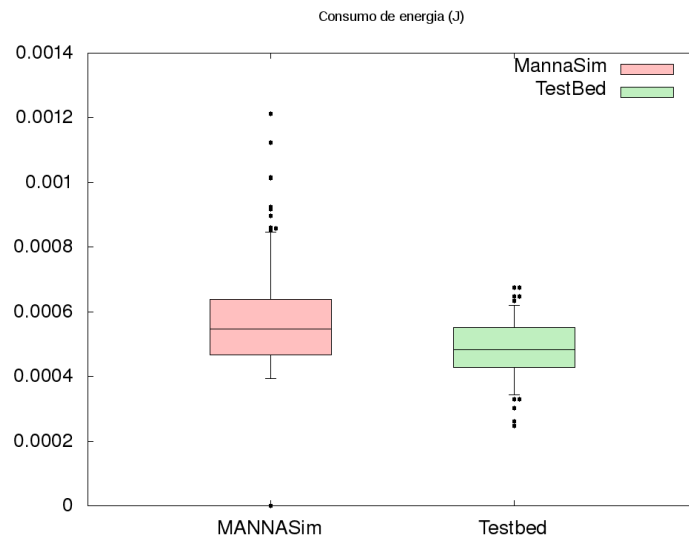


**Figura 4. Boxplot dos atrasos**

Realizando uma análise do boxplot gerado para os atrasos médios da rede, como mostra a Figura 4, levando em consideração uma margem de erro de  $10^{-3}$ , notamos que há uma convergência da mediana da métrica para ambos os cenários com um valor entre 0.013s e 0.017s, embora hajam valores fora da faixa e um desvio padrão no MANNASim maior do que o testbed.

Essa leve divergência se deve ao fato de o MANNASim estar utilizando o protocolo de roteamento DSDV - *Destination-Sequenced Distance Vector* - ao executar o experimento, isto é, um protocolo que prevê o uso de uma tabela para cada nó da rede contendo o nó destino, o próximo nó, o numero de saltos até chegar ao destino e outras informações. Para que opere de maneira correta, o DSDV exige que essa tabela seja atualizada regularmente. Tal feito garante um delay bem menor se for comparado a outros protocolos existentes.

- **Consumo energético da rede:**



**Figura 5. Boxplot dos consumos energéticos**



Realizando uma análise do boxplot gerado para os consumos médios de energia da rede, como mostra a Figura 5, levando em consideração uma margem de erro de  $10^{-5}$ , notamos uma aproximação das medianas da faixa de valores compreendida entre 0.025V a 0.029V. Os desvios padrão estão levemente semelhantes, com valores de ambas as partes fora da faixa.

A divergência encontrada nos dados se deve também ao uso do protocolo de roteamento DSDV que, conforme detalhado no tópico anterior, realiza uma atualização constante de suas tabelas de roteamento. Para tal, há um consumo de energia um pouco maior do que um protocolo utilizado em RSSFs que otimiza ao máximo o fator energético.

- **Número de mensagens enviadas:**

**Tabela 2. Número de mensagens enviadas para o ponto de acesso**

Nó	TestBed	MANNASim
PA	0	0
1	289	299
5	289	299
6	289	299
8	289	299
9	289	299

O número de mensagens enviadas pelos nós da rede para o ponto de acesso, de acordo com a tabela 2, considerando os 5 nós apresentados na seção anterior, apresentou uma diferença de 10 mensagens entre as plataformas consideradas nesse artigo.

Nota-se que há uma divergência no número de mensagens enviadas pelo fato de haver um atraso entre o instante em que inicia-se a simulação no ambiente remoto do testbed e o instante em que os nós finalizam seu processo de inicialização. Esse intervalo de tempo de alguns segundos certamente não é levado em conta ao se definir a cota de tempo do experimento no testbed.

## **6. Conclusão e Trabalhos Futuros**

A validação de um software é uma etapa fundamental em seu ciclo de vida, sendo imprescindível para sua aceitação, seja no mercado ou no ramo acadêmico. Viu-se por meio deste trabalho, que o MANNASim é um poderoso aliado ao lado de pesquisadores de RSSFs em todo o mundo e que pode ser utilizado como um simulador confiável, pois se mostrou compatível com a realidade.

Como trabalhos futuros, pretende-se realizar testes massivos com o MANNASim, de modo a identificar possíveis falhas e aprimorá-lo ainda mais. Além disso, é preciso atualizar os algoritmos de roteamento de RSSFs no MANNASim, para que os mais recentes e adotados em redes reais sejam adotados.

## Referências

- Akyildiz, I., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer Networks*, 38(4):393 – 422.
- Araújo, H. S., de Castro, W. L. T., and Filho, R. H. (2009). Simulação em rssf para protocolos de roteamento usando uma abordagem geocast.
- Braga, T. R. M. et al. (2004). Mannasim: Um arcabouço para simulação de redes de sensores sem fio.
- Doddavenkatappa, M., Chan, M. C., and Ananda, A. L. (2012). *Indriya: A Low-Cost, 3D Wireless Sensor Network Testbed*, pages 302–316. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Eclipse Foundation. Eclipse. <https://eclipse.org/>. [Acessado em 06 de Junho de 2016].
- Eriksson, J. et al. (2008). Demo abstract: Mspsim – an extensible simulator for msp430-equipped sensor boards.
- Levis, P. and Lee, N. (2003). *TOSSIM: A Simulator for TinyOS Networks*.
- Loureiro, A. A. et al. (2003). *Redes de Sensores Sem Fio*. Simpósio Brasileiro de Computação, Natal, RN.
- Manna Research Group et al. (2002). Mannasim framework. <http://www.mannasim.dcc.ufmg.br/acknowledgment.htm>. [Acessado em 06 de Junho de 2016].
- Pediaditakis, D., Tselishchev, Y., and Boulis, A. (2010). Performance and scalability evaluation of the castalia wireless sensor network simulator. In *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques, SIMUTools '10*, pages 53:1–53:6, ICST, Brussels, Belgium, Belgium. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
- Polley, J., Blazakis, D., McGee, J., Rusk, D., and Baras, J. S. (2004). Atemu: a fine-grained sensor network simulator. In *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pages 145–152.
- Raychaudhuri, D., Seskar, I., Ott, M., Ganu, S., Ramachandran, K., Kremo, H., Siracusa, R., Liu, H., and Singh, M. (2005). Overview of the orbit radio grid testbed for evaluation of next-generation wireless network protocols. In *IEEE Wireless Communications and Networking Conference, 2005*, volume 3, pages 1664–1669 Vol. 3.
- Sehgal, A. (2013). *Using the Contiki Cooja Simulator*. Jacobs University Bremen Campus Ring 1, 28759. Bremen, Germany.
- Titzer, B. L., Lee, D. K., and Palsberg, J. (2005). Avrora: Scalable sensor network simulation with precise timing. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks, IPSN '05*, Piscataway, NJ, USA. IEEE Press.
- UC Berkeley et al. (1995). The ns manual. <http://www.isi.edu/nsnam/ns/>. [Acessado em 06 de Junho de 2016].

Varga, A. (1993). The omnet++ network simulator framework. <https://omnetpp.org/>. [Acessado em 06 de Junho de 2016].

Werner-Allen, G., Swieskowski, P., and Welsh, M. (2005). Motelab: a wireless sensor network testbed. In *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, pages 483–488.