

UNIVERSIDADE ESTADUAL DE CAMPINAS
Instituto de Computação

Avaliação do Consumo de Energia em Rede WSN em Ambiente
Simulado

MO809A Tópicos de Computação Distribuída

Autor:
Eduardo Seiti Ito

Professor:
Leandro Villas



UNICAMP

Campinas - SP
Junho de 2019

Sumário

1	Introdução	1
2	Metodologia	2
2.1	NS3	3
2.2	OMNETPP	6
3	Análise dos Resultados	6
3.1	TC1, Variação da distância até o Rx com Wifi padrão IEEE802.11b	6
3.2	TC2, Variação da distância até o Rx com Wifi padrão IEEE802.11b com phyMode=DsssRate11Mbs	7
3.3	TC3, Variação da distância até o Rx com Wifi padrão IEEE802.11a com phyMode=OfdmRate12Mbps	9
3.4	TC4: Verificar o nível de consumo (em Joules) baseado no tempo de si- mulação (em segundos)	10
4	Conclusão	11

1 Introdução

Em Redes WSN (Wireless Sensor Network) se requer uma duração prolongada das baterias de sensores de forma que há um interesse em entender o comportamento do consumo de energia dos mesmos.

De acordo com o estudo realizado por Transviña-Moreno et al [2], o consumo de energia em dispositivos WiFi se deve à máquina de estado finito (FSM Finite State Machine) que descreve os três estados de um nó sensor.

- Estado de Inicialização: Inicialização do hardware (aquecimento do oscilador e periféricos, inicialização de variáveis, etc.)
- Estado ativo. Tarefa de sensoriamento como por exemplo transmissão de dados. Enquanto em operação, o nó sensor habilita o conversor AD periférico e o desabilita na conclusão da mesma. No caso da difusão da mensagem, o módulo de rádio necessitará se juntar à rede, reportar os dados e fechar a conexão de energia para evitar o consumo desnecessário de energia.
- Estado de Baixa Potência. O módulo de rádio será desligado, o microcontrolador será mantido em modo de dormência e todos os periféricos serão desabilitados ou ligados à um estado de menor nível de consumo.

A figura abaixo ilustra o estado FSM:

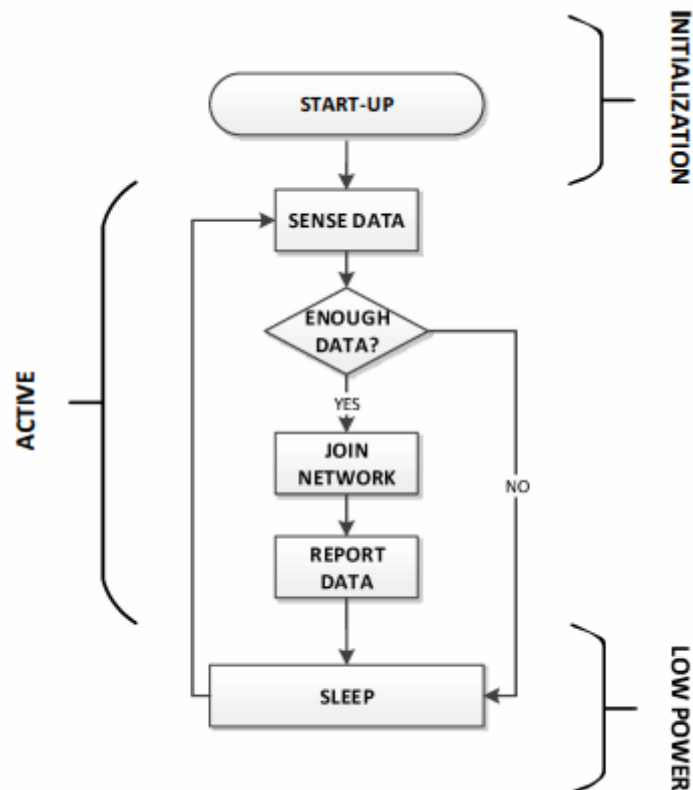


Figura 1: Máquina de Estado Finito (FSM)

O objetivo deste trabalho é verificar por meio de ambiente simulado o comportamento da energia em um nó sensor além dos estados FSM. Se há outros fatores que podem também influenciar o consumo de energia tais como a distância entre os sensores, versões de protocolos, ou mesmo se um nó comunica com mais de um nó. E também investigar quais as consequências de se aumentar a distância, se pode haver perda de pacotes. Finalmente, concluir com alguma percepção sobre a saída dos simuladores e alguns resultados de análise dos logs.

É importante chegar à alguma conclusão sobre o consumo de energia, pois sempre será requerido dimensionamento da duração da bateria em projetos WSN. Sensores normalmente operam de forma autônoma, contando apenas com a bateria, eventualmente com uma suplementação de carga com um coletor externo, e.g. placas fotovoltaicas.

Finalmente, será explicado a metodologia utilizada, análise dos resultados e uma conclusão final.

2 Metodologia

Nesta seção apresentamos os principais conceitos e técnicas que serão abordadas para a realização deste Trabalho Prático.

Todo o material produzido, shell script, programa python e .cc estão armazenados no GitHub¹

Os simuladores mais conhecidos são os listados abaixo: NS3², OMNET++³, WSNET⁴, SENSE⁵

Será priorizado neste trabalho prático, os simuladores NS3 e OMNET++, pelo fato de já terem sido apreciados em classe, como mostrados nos subcapítulos abaixo.

¹<https://github.com/edbkei/mo809a-trabalhopratico/>

²<https://www.nsnam.org/>

³<https://omnetpp.org/>

⁴<http://wsnet.gforge.inria.fr/>

⁵<https://www.ita.cs.rpi.edu/>

2.1 NS3

O módulo de energia do NS3 é baseado no Energy Model⁶, que é o resultado do modelo de energia proposto por Taparello et all [1], a figura abaixo ilustra o diagrama de energia.

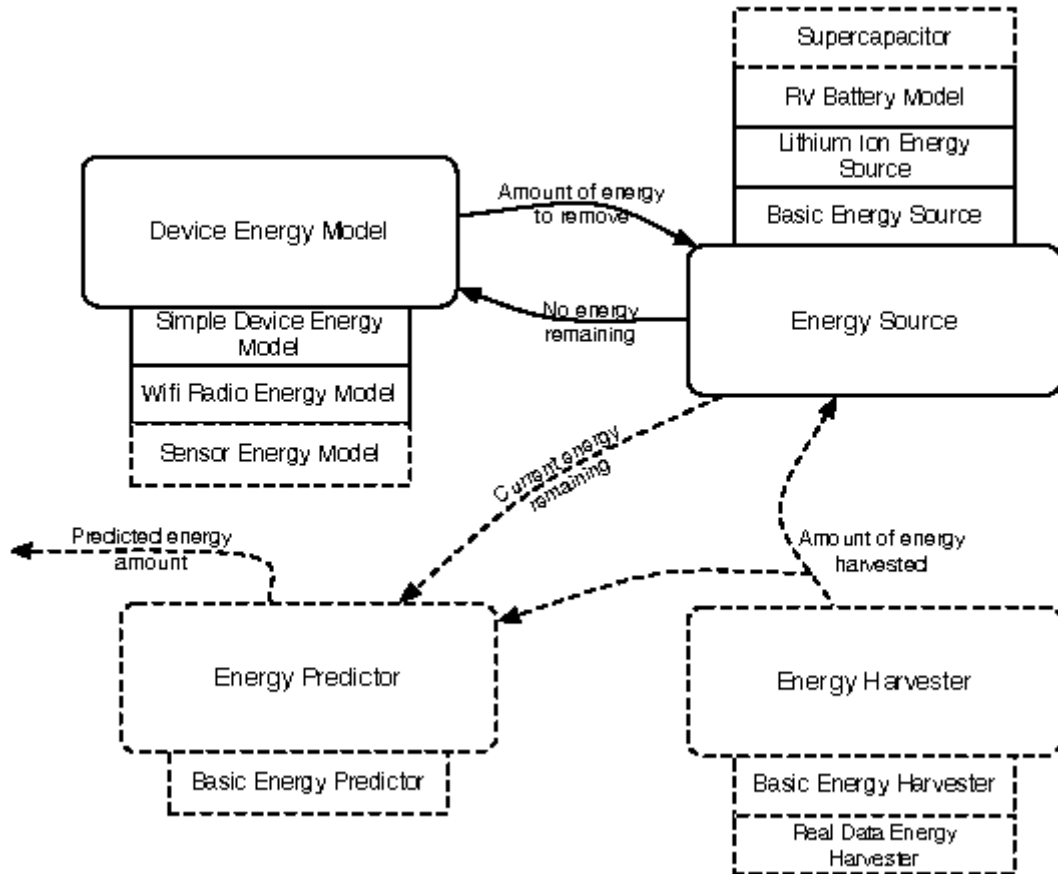


Figura 2: Modelo de Energia no NS3.

À cada dispositivo WSN um modelo de energia será conectado, no caso será o Wifi Radio Energy Model, cujo papel será administrar a energia gerada por uma a fonte de energia (Energy Source), no nosso caso será o Basic Energy Source. A fonte de energia receberá uma alimentação suplementar do Coletor de Energia (Energy Harvester), o modelo a ser utilizado é o Basic Energy harvester. Um exemplo seria placa de energia foto-voltaicas. O exemplo será baseado no programa exemplo do próprio NS3, energy-model-with-harvesting-example. cc.

A instalação do NS3 foi realizada em uma máquina virtual Ubuntu 18.04.2 LTS e os programas de energia estão dispostas da seguinte forma:

- Diretório /repos/ns-3-allinone/ns-3-dev/scratch
 - runtrab.sh, shell script para rodar de forma interativa o trabprat.cc e o programa stat-trab.py. Este script permite a escolha do número de interações da distância, que é dado em metros. Um parâmetro de entrada é o tempo de simulação em segundos. Este script contém o comando waf para rodar e compilar o programa trabprat.cc.

⁶<https://www.nsnam.org> -i Documents -i development-tree -i Doxygen -i Modules -i Energy Models

- `trabprat.cc`, programa em C++, baseado no exemplo dado do NS3 `energy-model-with-harvesting-example.cc`. Programa principal para geração do log de energia (em Joules), baseado em distâncias (metros). Outras características do sensor podem ser definidas, como por exemplo: o protocolo Wifi IEEE 802.11a, b, número de nós, adição do coletor adicional de energia (harvester).
- `stattrib.py`, script em Python para obtenção da estatística baseados nos logs gerados pelo `trabprat.cc`. Basicamente a entrada do log com pacotes enviados, recebidos, energia consumida pelo nó são plotados em um gráfico. Exemplo de um log do NS3 é mostrado no GitHub.

O programa `trabprat.cc` vai simular 2 nós Wifi, onde o nó de origem (Tx) envia pacotes UDP para o nó de destino. Como cada nó se conecta à um módulo de energia e este à um coletor suplementar de energia, de forma que será suficiente a coleta log da energia consumida de apenas um nó, e definiremos o nó Rx, como o coletor do log de energia. Isto é realizado através da sintaxe `DynamicCast<BasicEnergySource>(sources.Get(1))` e `TraceconnectWithoutContext`. Outras sintaxes são incluídas para a conexão com o coletor suplementar (harvester).

Depois de uma investigação preliminar, foram observadas que o consumo do dispositivo WiFi, estão situadas por volta de 0.82 J/s. Portanto uma observação de duração de 100 s, necessitaria pelo menos de uma carga inicial de 82 J, mas vamos considerar 100 J definido no parâmetro `BasicEnergySourceInitialEnergyJ`. Se não for feito este dimensionamento, haverá perda de pacotes por falta de energia durante a observação. No entanto, há uma suplementação de energia importante proveniente do harvester, que dá mais autonomia para um sensor. Isto pode ser observado pelo script `test1.sh`.

Quaisquer mudanças no comportamento do sensor, tais como mudança no padrão IEEE802.11x, distância até o Rx, tempo de simulação, etc devem ser realizados por parâmetros externos ou por mudança no próprio código `.cc`. Nem tudo está parametrizado por restrição de tempo do curso MO809A. Os itens abaixo descrevem o que são os parâmetros variáveis e fixos.

- Parâmetros variáveis utilizados:
 - `simTime`, tempo de simulação em segundos.
 - `distanceToRx`, distância até o Rx em metros.
 - outros parâmetros existentes eventualmente podem ser utilizados. `phyMode` (Wifi Physical Mode), `Prss` (Intended primary RSS (dBm)), `PacketSize` (tamanho do pacote), `numPackets` (número de pacotes para serem enviados), `startTime` (momento para o início da simulação)
- Parâmetros fixos utilizados (modificação apenas no código fonte `.cc`)
 - Número de nós. Comando `create()`.
 - modificação do padrão Wifi. `SetStandard()`.
 - posicionamento do dispositivo Wifi. `positionAlloc`.
 - Corrente em Ampere do Tx e Rx. `radioEnergyhelper.Set()`.
 - Mudança do número IP. `ipv4.SetBase()`.

Para animação da rede de sensores, basta colocar 2 linhas de comando no código fonte `trabprat.cc`.

```
include "ns3/netanim-module.h"
```

```
AnimationInterface anim("arquivo.xml"), antes do Simulator::Run()
```

Fazer a carga do arquivo após o comando `./NetAnim`

Como são simulações específicas de dados de energia e isso não é bem representado no modo gráfico com NetAnim, nem com opção `-vis` no comando `./waf`, prefere-se utilizar a saída dos dados de energia do programa `trabprat.cc` e plotar o gráfico com a opção `matplotlib` do Python.

Para avaliar o comportamento de energia no nó sensor Rx, serão propostos os seguintes casos de teste.

- TC1. Verificar o nível de consumo (em Joules) variando a distância até o Rx (em metros). Pré-condição: Valor padrão para o IEEE802.11b tais como `phyMode DsssRate1Mbs`, `Prss -80 dBm`, `PacketSize 200`, `RxGain -10 dBm`, `TxGain -80 (Prss)+81(offset)`. Pode ser utilizado diretamente o script `trabprat.sh` com parâmetros de distância até o Rx, número de interações e tempo de simulação. Ou pode ser utilizado o `test4.sh` com edição desses parâmetros para o uso do próprio `trabprat.sh`. A distância até o Rx vai variar desde 100m, 200m, até 1400m, onde deve ocorrer a maior perda de pacotes.
- TC2. Verificar o nível de consumo (em Joules) variando a distância até o Rx (em metros). O mesmo que TC1, mas aumentando o taxa de transmissão, com `DsssRate11Mbs`. Seria o mesmo que TC1 com `phyMode=DsssRate11Mbs`, i.e. tx de 11 Mb/s.
- TC3. Verificar o nível de consumo (em Joules) variando a distância até o Rx (em metros). Utilizando o padrão IEEE802.11a `phyMode=OfdmRate12Mbps`. Versões mais antigas do padrão WiFi, pode ter uma diferença considerável no consumo de energia. Configuração pode ser encontrada na referência do ns3 ⁷
- TC4. Verificar o nível de consumo (em Joules) baseado no tempo de simulação. A intenção é verificar se o coletor suplementar de energia (harvester) contribui com uma carga razoavelmente boa em relação ao consumo do nó Rx. Será utilizado basicamente o script `test1.sh` com uma duração de 120 segundos e distância de 10 metros fixo. Foi utilizado o padrão Wifi IEEE802.11b.

⁷<https://www.nsnam.org/bugzilla/attachment.cgi?id=2230>

2.2 OMNETPP

O módulo de energia do OMNETPP será baseado no inet-framework⁸

3 Análise dos Resultados

3.1 TC1, Variação da distância até o Rx com Wifi padrão IEEE802.11b

Os dados de energia na simulação com variação de distância até o Rx podem ser observados na tabela abaixo.

Tabela de Energia WiFi IEEE802.11b				
simTime	distanceToRx	pktReceived	pktSent	totalEnergy
100	100	100	100	81.7252
100	200	100	100	81.7252
100	300	100	100	81.7252
100	400	100	100	81.7252
100	500	100	100	81.7252
100	600	100	100	81.7252
100	700	100	100	81.7252
100	800	100	100	81.7252
100	900	100	100	81.7252
100	100	98	100	81.7252
100	1100	80	100	81.7252
100	1200	56	100	81.7252
100	1300	10	100	81.7252
100	1400	0	100	81.7252

Observe que não há variação do consumo de energia do nó Rx. Sempre 81.7252J a cada 100s.

O teste hipótese

$$H_o : \mu = 81.7252$$

utilizando o teste de t-student seria aceito porque o desvio padrão seria s=0, onde

$$s = \sqrt{\frac{\sum (x_i - 81.7252)^2}{n - 1}}$$

O gráfico de energia ficaria da seguinte forma:

⁸<https://github.com/inet-framework/inet-showcases/tree/master/wireless/power>

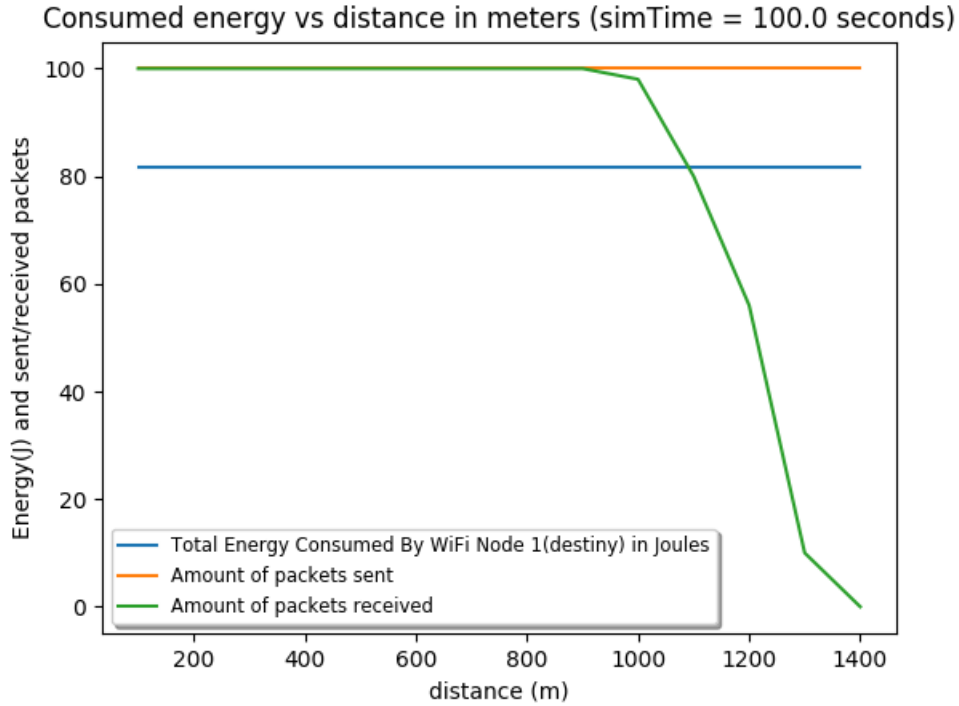


Figura 3: Gráfico de energia vs distância até o Rx, IEEE802.11b (DsssRate1Mbps)

A cada passo de 100m com tempo de simulação de 100s, não há diferença de consumo de energia do nó Rx, sempre a 81.7252J. O que se percebe é a perda de pacotes com a distância a partir de 800m, possivelmente devido à degradação do sinal Wifi e a potência na recepção seja menor do que é estabelecido pelo Rx Sensibility, no nó de destino.

3.2 TC2, Variação da distância até o Rx com Wifi padrão IEEE802.11b com phyMode=DsssRate11Mbps

Os dados de energia na simulação com variação de distância até o Rx podem ser observados na tabela abaixo.

Tabela de Energia WiFi IEEE802.11b, phyMode=DsssRate11Mbps				
simTime	distanceToRx	pktReceived	pktSent	totalEnergy
100	100	100	100	81.8711
100	200	100	100	81.8711
100	300	99	100	81.8711
100	400	13	100	81.8711
100	500	0	100	81.8711
100	600	0	100	81.8711
100	700	0	100	81.8711
100	800	0	100	81.8711
100	900	0	100	81.8711
100	100	0	100	81.8711
100	1100	0	100	81.8711
100	1200	0	100	81.8711
100	1300	0	100	81.8711
100	1400	0	100	81.8711

Observe que não há variação do consumo de energia do nó Rx. Sempre 81.8711J a cada 100s.

O teste hipótese

$$H_0 : \mu = 81.7252$$

utilizando o teste de t-student seria aceito porque o desvio padrão seria $s=0$, onde

$$s = \sqrt{\frac{\sum (x_i - 81.8711)^2}{n - 1}}$$

Mas se for comparar as médias de TC1 e deste TC2, a hipótese

$$H_0 : \mu_1 = \mu_2$$

seria rejeitada porque os desvios padrões $s_1 = s_2 = 0$ e o estimador t-student tenderia para o infinito.

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\left(\frac{(n-1)s_1^2 + (n-1)s_2^2}{2n-2}\right)\left(\frac{2}{n}\right)}}$$

O gráfico de energia ficaria da seguinte forma:

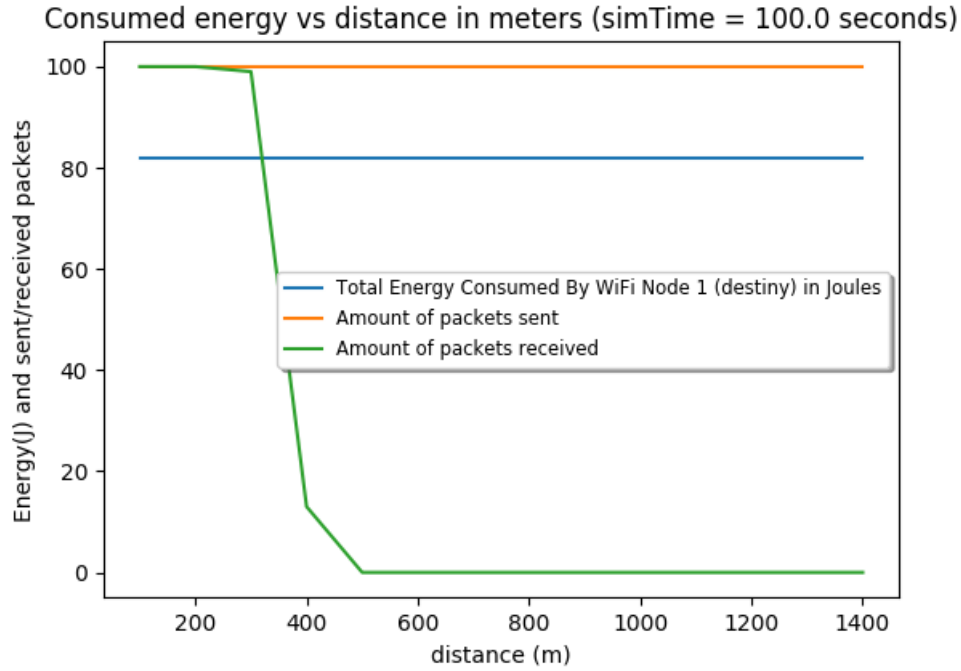


Figura 4: Gráfico de energia vs distância até o Rx, IEEE802.11b (DsssRate11Mbps)

A cada passo de 100m com tempo de simulação de 100s, não há diferença de consumo de energia do nó Rx, sempre a 81.8711J. O que se percebe é a perda de pacotes com a distância a partir de 200m, possivelmente devido à degradação do sinal Wifi e a potência na recepção seja menor do que é estabelecido pelo Rx Sensibility, no nó de destino.

3.3 TC3, Variação da distância até o Rx com Wifi padrão IEEE802.11a com phyMode=OfdmRate12Mbps

Os dados de energia na simulação com variação de distância até o Rx podem ser observados na tabela abaixo.

Tabela de Energia WiFi IEEE802.11a, phyMode=OfdmRate12Mbps				
simTime	distanceToRx	pktReceived	pktSent	totalEnergy
100	100	0	100	81.8851
100	200	0	100	81.9
100	300	0	100	81.9
100	400	0	100	81.9
100	500	0	100	81.9
100	600	0	100	81.9
100	700	0	100	81.9
100	800	0	100	81.9
100	900	0	100	81.9
100	1000	0	100	81.9
100	1100	0	100	81.9
100	1200	0	100	81.9
100	1300	0	100	81.9
100	1400	0	100	81.9

Observe que quase não há variação do consumo de energia do nó Rx. Sempre 81.9J a cada 100s. Um consumo um pouco superior ao IEEE802.11b.

O gráfico de energia ficaria da seguinte forma:

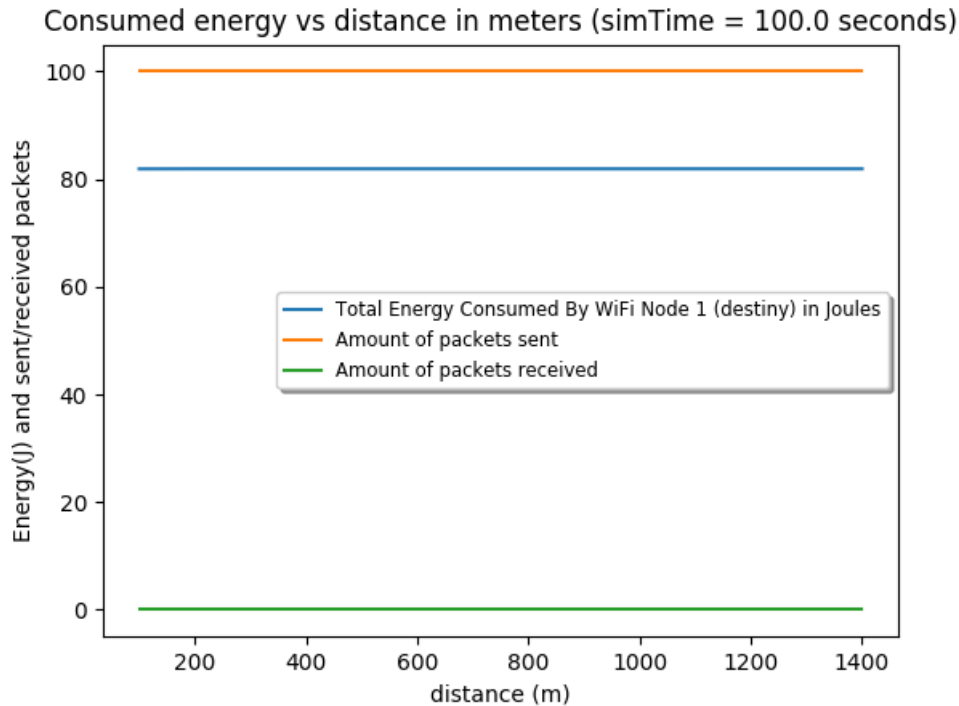


Figura 5: Gráfico de energia vs distância até o Rx, IEEE802.11a

Observe que além de o consumo manter constante à 81.9 J em 100 segundos. O nó Rx não recebe nenhum pacote do nó Tx. Mesmo dobrando o valor da amperagem em Rx (RxCurrentRxA), nenhum pacote é recebido. Isso merece uma investigação separada.

3.4 TC4: Verificar o nível de consumo (em Joules) baseado no tempo de simulação (em segundos)

Os dados de energia na simulação com tempo de simulação de 120s e com distância até o Rx fixa de 10 metros podem ser observados na tabela abaixo.

Tabela de Energia consumida e colhida	
totalEnergyConsumedByRadio(J)	totalEnergyHarvestedByHarvester(J)
95.6185	5.66049

Observe que a energia colhida representa apenas 6 porcentos da energia consumida.

O gráfico de energia consumida e colhida ficaria da seguinte forma:

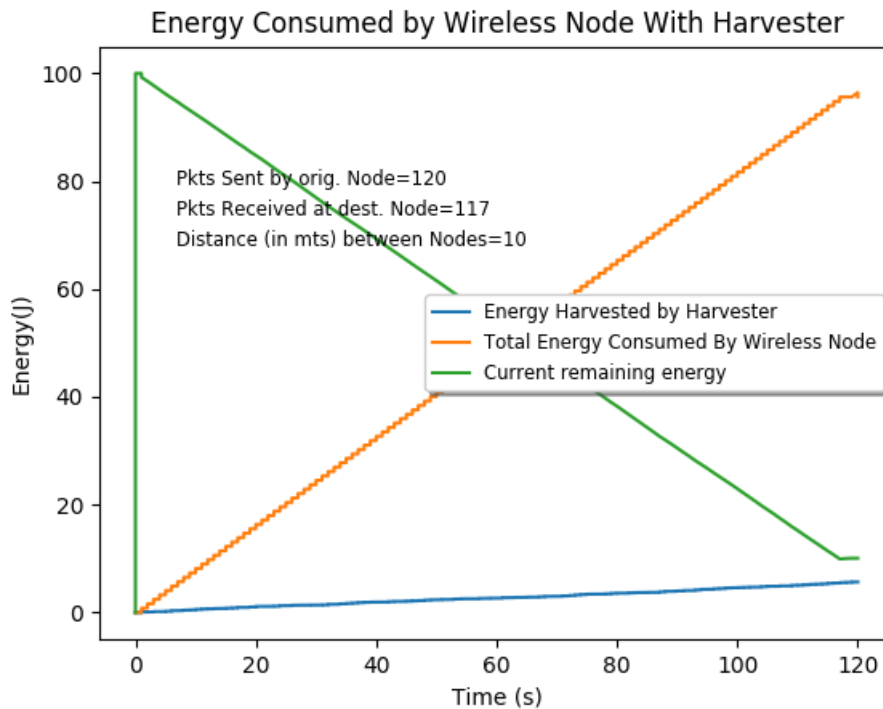


Figura 6: Gráfico de energia vs tempo de simulação de 120s, IEEE802.11b

O gráfico mostra que o nó Rx tem pouca autonomia de energia com apenas a carga fornecida pelo coletor suplementar (harvester). A energia consumida é muito maior e a consequência é a perda de pacote. Portanto, a maior autonomia seria obtida por conta da carga inicial.

4 Conclusão

De acordo com os resultados obtidos pelo simulador NS3, não há diferença do consumo de energia se considerar a variação de distância até o Rx. Haverá diferença de consumo se variar alguma característica do padrão Wifi, como o physical mode (e.g. se utilizar DsssRate1Mbs ou DsssRate11Mbs) ou variar a versão do protocolo Wifi como por exemplo IEEE802.11a ou IEEE802.11b.

A perda de pacotes UDP varia de acordo com a distância até o Rx e também por alguma mudança na característica do padrão Wifi.

Um coletor de energia suplementar (harvester) contribui com muito pouco na alimentação do nó Rx, apenas 6 por cento é obtido. Uma carga inicial maior é requerida para uma maior autonomia.

Um trabalho futuro é recomendado para investigar o consumo de energia com todas as características de um sensor, não apenas aquele baseado em IEEE802.11x, mas outros protocolos também seriam de grande valia, como o LoRaWan.

Referências

- [1] C. Tapparello, H. Ayatollahi, and W. Heinzelman. Energy harvesting framework for network simulator 3 (ns-3). 11 2014.
- [2] C. Trasviña-Moreno, A., n. Asensio, R. Casas, R. Blasco, and l. Marco. Wifi sensor networks: a estudy of energy consumption. *ResearchGate*, Feb. 2014.