

Renan Trindade Paim

**Redes de Sensores Sem Fio (RSSF) para  
Monitoramento por Geolocalização para  
Pecuária: Estudo de Caso Empregando o  
Simulador NS-2**

Alegrete, RS

14 de dezembro de 2017

Renan Trindade Paim

**Redes de Sensores Sem Fio (RSSF) para Monitoramento  
por Geolocalização para Pecuária: Estudo de Caso  
Empregando o Simulador NS-2**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Telecomunicações, Área de Concentração em Eletrônica, da Universidade Federal do Pampa (Unipampa, RS), como requisito parcial para obtenção do grau de **Bacharel em Engenharia de Telecomunicações**.

Universidade Federal do Pampa – Unipampa  
Curso de Engenharia de Telecomunicações

Orientador: Prof. Me. Paulo César Comassetto de Aguirre

Alegrete, RS  
14 de dezembro de 2017

Ficha catalográfica elaborada automaticamente com os dados fornecidos  
pelo(a) autor(a) através do Módulo de Biblioteca do  
Sistema GURI (Gestão Unificada de Recursos Institucionais) .

P133r Paim, Renan

Redes de Sensores Sem Fio (RSSF) para Monitoramento por Geolocalização para Pecuária: Estudo de Caso Empregando o Simulador NS-2. / Renan Paim.

55 p.

Trabalho de Conclusão de Curso(Graduação)-- Universidade Federal do Pampa, ENGENHARIA DE TELECOMUNICAÇÕES, 2017.

"Orientação: Paulo César de Aguirre".

1. Universidade Federal do Pampa. 2. Redes de Sensores Sem Fio. 3. Network Simulator (NS-2). 4. Redes de Sensores Sem Fio (RSSF) para Monitoramento por Geolocalização para Pecuária: Estudo de Caso Empregando o Simulador NS-2. I. Título.

**RENAN TRINDADE PAIM**

**REDE DE SENSORES SEM FIO (RSSF) PARA MONITORAMENTO POR GEOLOCALIZAÇÃO PARA  
PECUÁRIA: ESTUDO DE CASO UTILIZANDO O SIMULADOR NS-2**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Telecomunicações da Universidade Federal do Pampa, como requisito parcial para obtenção do título de Bacharel em Engenharia de Telecomunicações.

Área de Concentração: Eletrônica

Trabalho de Conclusão de Curso defendido e aprovado em: 29 de Novembro de 2017.

Banca examinadora:



---

Prof. Me. Paulo César Comassetto de Aguirre  
Orientador



---

Prof. Me. Cristian Müller  
UNIPAMPA



---

Prof. Dr. Bruno Boessio Vizzotto  
UNIPAMPA

*Dedico este trabalho aos meus pais,  
como forma de reconhecimento e agradecimento  
por todo apoio, amor e inspiração durante esses anos.*

# Agradecimentos

Agradeço a minha família, em especial aos meus pais, Jaqueline e Nelcy, por todo apoio e incentivo durante essa caminhada, pois sem eles nada seria possível. Agraço também aos meus irmãos, Bruna e Dimitri, pelo carinho e companheirismo durante todos esses anos.

Agradeço a minha namorada, Carolina, por todo amor, carinho, apoio e incentivo durante todos esses anos, também pelas conversas, risadas e companheirismo.

Agradeço também ao meu orientador Prof. Me. Paulo César Comassetto de Aguirre e ao Prof. Me Cristian Müller por embarcarem nessa comigo, pois sem a ajuda de vocês este trabalho não seria realizado, obrigado pelos ensinamentos passados, pela ajuda prestada, pelas diversas conversas e reuniões, enfim, obrigado por todo apoio.

Por fim, aos colegas de laboratório (GAMA), em especial, o Anderson e o Alian, por toda ajuda prestada durante a elaboração deste trabalho, pelas dúvidas sanadas, pelas explicações da linguagem de programação, pelas companhias nas madrugadas, enfim, pelo companheirismo ao longo do tempo que eu estive no grupo GAMA.

*“O que for teu desejo, assim será tua vontade.  
O que for tua vontade, assim serão teus atos.  
O que forem teus atos, assim será teu destino”.*  
*(Deepak Chopra)*

# Resumo

Este trabalho apresenta um estudo sobre redes de sensores sem fio (RSSF) para monitoramento por geolocalização para pecuária empregando o software de simulação *Network Simulator 2* (NS-2). Visando analisar a duração da RSSF efetuou-se a análise do impacto do protocolo de roteamento empregado na rede e do número de nós da RSSF, além do intervalo de transmissão de pacotes. Como estudos de caso utilizaram-se redes de sensores sem fio compostas por um concentrador e quinze, trinta, e quarenta e cinco nós sensores, além de intervalos de transmissão de cinco e dez minutos. Foram analisados três protocolos de roteamento *unicast*, são eles: *Ad Hoc On Demand Distance Vector Routing* (AODV), *Dynamic Source Routing* (DSR) e *Destination Sequenced Distance Vector* (DSDV). Para a avaliação dos protocolos foram comparadas as métricas de desempenho, como o *overhead*, *throughput*, relação de entrega de pacotes (PDR), atraso médio da rede e consumo médio de energia para um hardware composto por um Arduino Uno, um transceptor nRF905 e um módulo GPS. Os resultados obtidos indicam que o protocolo DSR apresentou um melhor resultado para o cenário em que o tempo entre as transmissões é menor, e o protocolo AODV apresentou um melhor desempenho para um maior tempo entre as transmissões, sendo que em nenhum dos cenários o protocolo DSDV apresentou melhor desempenho que os demais protocolos.

**Palavras-chaves:** Redes Sensores Sem Fio, Network Simulator (NS-2), AODV, DSR, DSDV.



# Abstract

This work presents a study on wireless sensor networks (WSNs) for geolocation monitoring for livestock using the Network Simulator 2 (NS-2) simulation software. In order to verify the duration of the WSN, it is analyzed the impact of the routing protocol employed in the network and the number of nodes of the WSN network, in addition to the packet transmission interval. As case studies, wireless sensor networks composed of one concentrator and fifteen, thirty, and forty-five sensor nodes were used, in addition to five minutes and ten minutes transmission intervals. Three unicast routing protocols were analyzed: Ad Hoc On-Demand Distance Vector Routing (AODV), Dynamic Source Routing (DSR) and Destination Sequenced Distance Vector (DSDV). Performance metrics such as overhead, throughput, packet delivery ratio (PDR), average network delay and average power consumption were compared to evaluate the impact of using different protocols. Each simulated node consists of an Arduino Uno, a nRF905 transceiver, and one GPS module. The obtained results indicate that the DSR protocol provides a longer RSSF lifetime in the scenario with transmission interval of five minutes, and the AODV protocol presented a better performance for transmission interval of ten minutes, and in none of the scenarios, the DSDV protocol presented better performance than the other protocols.

**Key-words:** Wireless Sensor Network, Network Simulator (NS-2), AODV, DSR, DSDV.

# Lista de ilustrações

Figura 1 – Em (a) transmissão de A para B em um único salto e em (b) transmissão de A para B em dois saltos. . . . .	15
Figura 2 – Visão de como as RSSFs podem interagir com outras redes. . . . .	16
Figura 3 – RSSF com múltiplos concentradores. . . . .	17
Figura 4 – Diagrama de blocos do nó sensor de uma RSSF. . . . .	18
Figura 5 – Pilha de protocolos das redes de sensores sem fio. . . . .	21
Figura 6 – Exemplo da Rede Sensor Sem Fio na Prática. . . . .	29
Figura 7 – Arduino Uno. . . . .	31
Figura 8 – nRF 905. . . . .	33
Figura 9 – Módulo GPS. Adaptado de (ELECTRONICS, 2012). . . . .	34
Figura 10 – Componentes de software do simulador NS-2. . . . .	36
Figura 11 – Ilustração do processo de criação e resultados da simulação. . . . .	37
Figura 12 – Cenário Físico da RSSF. . . . .	39
Figura 13 – Cenário Físico da RSSF. . . . .	40
Figura 14 – Energia restante em cada nó da rede. . . . .	44
Figura 15 – Energia Consumida x Nó (Protocolo AODV). . . . .	44
Figura 16 – Energia Consumida x Nó (Protocolo DSDV). . . . .	45
Figura 17 – Energia Consumida x Nó (Protocolo DSR). . . . .	45
Figura 18 – Gráfico Vida Útil vs Tempo de Atualização do Módulo GPS. . . . .	47
Figura 19 – Energia Consumida x Nó (Protocolo AODV). . . . .	47
Figura 20 – Energia Consumida x Nó (Protocolo DSDV). . . . .	48
Figura 21 – Energia Consumida x Nó (Protocolo DSR). . . . .	48

# Lista de tabelas

Tabela 1 – Especificações Técnicas. . . . .	32
Tabela 2 – Especificações Técnicas do nRF 905. . . . .	33
Tabela 3 – Especificações do Módulo GPS. . . . .	34
Tabela 4 – Definição dos pinos do módulo. Fonte: Adaptado de (ELECTRONICS, 2012) . . . . .	34
Tabela 5 – Configurações dos Parâmetros de Hardware. . . . .	38
Tabela 6 – Configurações da Topologia e Tráfego de Dados da Rede. . . . .	38
Tabela 7 – Parâmetros configuráveis do Modelo de Energia. . . . .	40
Tabela 8 – Valores utilizados nas simulações no Modelo de Energia. . . . .	41
Tabela 9 – Resultados do Primeiro Cenário . . . . .	43
Tabela 10 – Resultados do Segundo Cenário. . . . .	46
Tabela 11 – Resultados da Influencia da Quantidade de Nós. . . . .	49

# Sumário

<b>1</b>	<b>Introdução</b>	<b>13</b>
1.1	Justificativa	14
1.2	Objetivos	14
1.2.1	Objetivos Específicos	14
<b>2</b>	<b>Revisão Bibliográfica</b>	<b>15</b>
2.1	Definições de Termos	15
2.2	Fundamentos das RSSFs	15
2.2.1	Restrições das RSSFs	17
2.2.1.1	Tolerância a falhas	18
2.2.1.2	Restrições de Hardware	18
2.2.1.3	Custo de Produção	19
2.2.1.4	Manutenção da Topologia da Rede de Sensores	19
2.2.1.5	Consumo de Energia	19
2.3	Vantagens das RSSFs	19
2.3.1	Eficiência Energética	20
2.4	Camadas das RSSFs	20
2.5	Classificação dos Protocolos para RSSFs	21
2.5.1	Protocolos Reativos	22
2.5.2	Protocolos Proativos	22
2.5.3	Protocolos Híbridos	23
2.6	Protocolo de Roteamento AODV	23
2.6.1	Descoberta de Rotas	23
2.6.2	Manutenção de Rotas	24
2.7	Protocolo de Roteamento DSR	25
2.7.1	Descoberta de Rotas	26
2.7.2	Manutenção de Rotas	26
2.8	Protocolo de Roteamento DSDV	27
2.8.1	Descoberta de Rotas	27
2.8.2	Manutenção de Rotas	27
2.9	Comentários	28
<b>3</b>	<b>Metodologia</b>	<b>29</b>
3.1	Arquitetura da RSSF	29
3.2	Componentes Utilizados	30
3.2.1	Processador	30
3.2.2	Transceptor	31
3.2.3	Módulo GPS	33

3.3	Comentários	33
<b>4</b>	<b>Estudo de Caso</b>	<b>36</b>
4.1	Network Simulator 2	36
4.2	Ambiente e Cenários de Simulação	37
4.3	Configuração do Consumo de Energia	39
4.4	Métricas de Desempenho	41
4.5	Análise dos Resultados	42
4.5.1	Intervalo de Transmissão de 5 Minutos	42
4.5.2	Intervalo de Transmissão de 10 Minutos	46
4.6	Influência da Quantidade de Nós na RSSF	46
4.7	Comentários	50
<b>5</b>	<b>Conclusão</b>	<b>51</b>
5.1	Trabalhos Futuros	51
	<b>Referências</b>	<b>53</b>

# 1 Introdução

A pecuária é um dos principais destaques do agronegócio brasileiro no cenário mundial. O Brasil é o dono do segundo maior rebanho efetivo do mundo, com cerca de 212,3 milhões de cabeças de gado, atrás apenas da Índia (BRASIL, 2015a). No Brasil, em 2015, a produção pecuária movimentou cerca de 168,1 bilhões de reais, que evidencia a importância econômica e social da pecuária em nosso país. Além disso, desde 2004, assumiu a liderança nas exportações de carne, com um quinto da carne comercializada internacionalmente e vendas em mais de 180 países (BRASIL, 2015b). Com isto, vêm crescendo os investimentos em tecnologias que permitam ao pecuarista ter um melhor controle, seja de localização, sanidade ou segurança alimentar do rebanho, para que o país atenda às exigências dos mercados rigorosos e conquiste mais espaço no cenário mundial.

Atualmente, uma alternativa de realizar o controle animal é através de sistemas de identificação por brincos ou dispositivos internos, que utilizam a tecnologia de identificação por rádio frequência (RFID - *radio frequency identification*). Entretanto, a atual aquisição de dados nesses sistemas costumam depender da intervenção humana, pois utilizam um leitor de rádio frequência para tais fins, que ocorre durante as práticas de manejo, por exemplo, em mangueiras. Esse tipo de sistema, embora eficiente, não permite que aspectos relacionados à localização e comportamento animal sejam monitorados em tempo real. Novos sistemas, compostos por Redes de Sensores Sem Fio (RSSF), que utilizam pequenos dispositivos com capacidade de processamento, armazenamento e comunicação sem fio, podem permitir o monitoramento individual dos animais em tempo real.

As aplicações das RSSFs cresceram nos últimos anos, principalmente devido aos estudos e inovações tecnológicas na área. O princípio de uma rede de sensores sem fio é o uso de uma grande quantidade de nós sensores, sujeitos a falha, com interligação sem fio entre eles, com isso, os protocolos de comunicação e gerenciamento dessas redes devem ter capacidades de auto-organização. Esses nós sensores devem ser de baixo custo, baixo consumo de energia e pequenos no tamanho (SILVA, 2011).

Independentemente do tipo de projeto, os simuladores de redes são ferramentas computacionais que auxiliam nas escolhas de decisões, já que possibilitam realizar a previsão de um ambiente real através da simulação. Através dos simuladores, é possível analisar diferentes cenários, sistemas e equipamentos envolvidos no projeto. Isto permite colher informações sobre as variações e modificações que foram testadas, a ponto de conseguir dados que comprovem a viabilidade de uma rede, para a implantação de um projeto real.

## 1.1 Justificativa

Devido ao crescente número de roubo de gado (abigeato) na Região Oeste do Estado do Rio Grande do Sul, nos últimos anos, tornou-se necessário a criação de uma sistema capaz de ajudar a inibir ou diminuir este fato. O prejuízo com abigeato somente no Estado do Rio Grande do Sul superou 70,6 milhões de reais no ano de 2016 ([AGRICULTURA, 2016](#)).

A motivação deste trabalho ocorre na simulação de um sistema (rede de sensores sem fio) que seja capaz de ajudar a prevenir o abigeato através do monitoramento da localização do gado no pasto.

## 1.2 Objetivos

Simular uma RSSF para o monitoramento por geolocalização para pecuária, através do software de simulação *Network Simulator* v.2 (NS-2). As simulações foram efetuadas de modo a analisar diferentes configurações de RSSFs e o tempo de vida útil da rede.

### 1.2.1 Objetivos Específicos

- Aprender a utilizar o software NS-2 para simulações de redes de sensores sem fio.
- Simular redes de sensores sem fio através do NS-2 para diversos cenários.
- Analisar e verificar os resultados obtidos a partir da simulação.

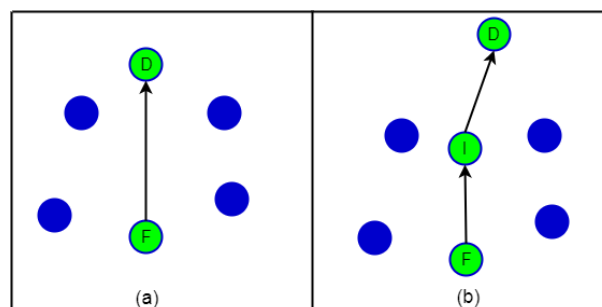
## 2 Revisão Bibliográfica

### 2.1 Definições de Termos

Para melhor entendimento sobre redes de sensores sem fio, serão definidos alguns termos conforme encontrado em (ZHAO; GUIBAS, 2004):

- **Nó sensor:** esse elemento é a unidade básica da rede de sensores. É composto de sensor(es), processador, módulo sem fio e fonte de alimentação. Esse elemento também é chamado simplesmente de nó.
- **Serviços de um nó:** são as funcionalidades oferecidas pelo nó, por exemplo, sincronização de tempo, localização do nó, requisição de leitura do sensor e funções de mobilidade.
- **Métricas de avaliação:** medida quantitativa que descreve o quão bem o sistema está executando suas funções, por exemplo, taxa de perda de pacotes, tempo de permanência da rede, consumo de energia e atraso médio da rede.
- **Transmissão multi-saltos e transmissão em um único salto:** uma transmissão entre dois nós é em um único salto se não transitar por nenhum outro nó da rede, como na Figura 1 (a). Já uma transmissão multi-saltos é aquela que se utiliza de outros nós da rede para traçar uma rota entre o emissor e o receptor, como na Figura 1 (b).

Figura 1 – Em (a) transmissão de A para B em um único salto e em (b) transmissão de A para B em dois saltos.



Fonte: Adaptado de (SILVA, 2011).

### 2.2 Fundamentos das RSSFs

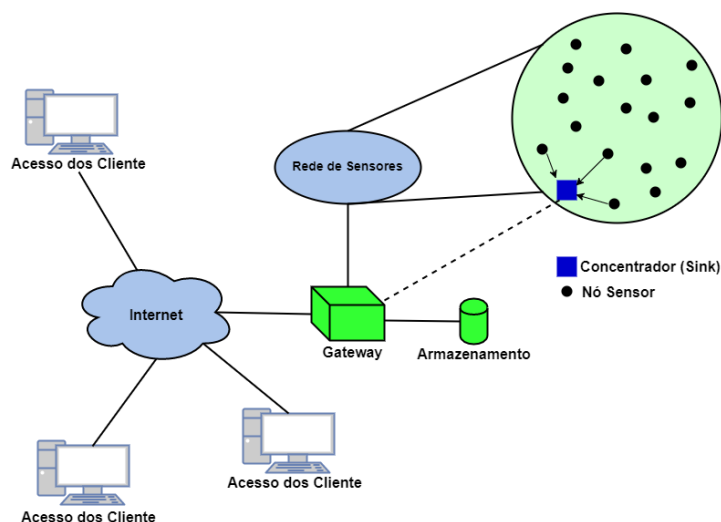
Uma RSSF pode ser caracterizada pelo uso de uma quantidade grande de nós sensores com a capacidade de comunicar a informação obtida da área monitorada através



de um enlace sem fio. O sensor é quem faz a monitoramento do fenômeno que está sendo analisado, repassando essa informação para o observador. Esta é transmitida para um nó concentrador, que pode localmente utilizar as informações coletadas, ou estar conectado a uma outra rede, geralmente a Internet, para oferecer acesso às informações coletadas para diversos usuários (AKYILDIZ et al., 2002).

A Figura 2 demonstra a interação entre as redes de sensores e a Internet. O elemento chamado *gateway* é utilizado para enviar comandos ou requisições para um determinado nó da rede de sensores e receber as informações adquiridas em campo, como os dados de estado da rede. Elementos de armazenamento, podem se encontrar junto ao *gateway* com a finalidade de guardar os registros das leituras e operações realizadas. O *gateway* deve possuir duas interfaces distintas de comunicação, uma para a rede de sensores sem fio e outra para a Internet (LIMA et al., 2009).

Figura 2 – Visão de como as RSSFs podem interagir com outras redes.



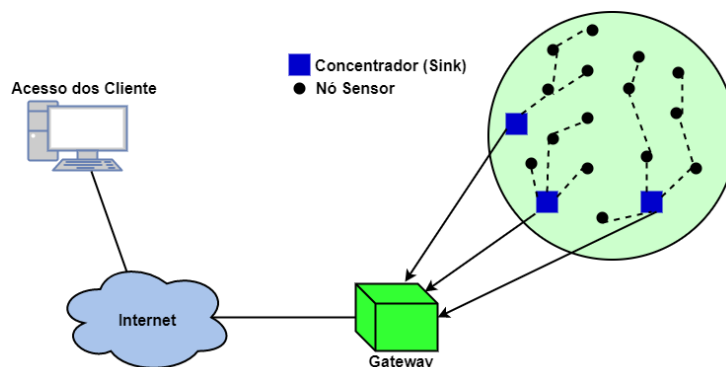
Fonte: Adaptado de (SILVA, 2011).

Outro elemento apresentado na Figura 2 é o concentrador ou *sink*. Tem como objetivo transferir as informações coletadas na área monitorada para o *gateway*. Em uma rede de sensores sem fio pode existir o problema de escalabilidade, pois aumentando o número de nós da RSSF, o concentrador pode ficar ineficiente em transferir as informações coletadas para o *gateway*. Além disso, por razões relacionadas ao controle de acesso ao meio (MAC – *Media Access Control*) e aspectos de roteamento, a performance de uma RSSF não pode ser considerada independente do seu tamanho (BURATTI, 2011).

Uma solução para este problema é a utilização de vários concentradores dentro de uma RSSF, como na Figura 3. Um número maior de concentradores fará com que diminua a probabilidade de isolamento dos nós, ou seja, que não consigam transferir seus dados coletados em campo. Entretanto, essa solução não é de trivial implementação, pois surge o problema de escolha do concentrador por parte do nó. No ponto de vista do protocolo,

essa decisão geralmente é tomada com base no atraso mínimo e número mínimo de saltos (BURATTI, 2011). Assim, o uso de múltiplos concentradores melhora o desempenho da RSSF dependendo da característica da rede. Entretanto, os protocolos para múltiplos concentradores são mais complexos.

Figura 3 – RSSF com múltiplos concentradores.



Fonte: Adaptado de (SILVA, 2011).

### 2.2.1 Restrições das RSSFs

As RSSFs possuem as mesmas dificuldades e problemas das redes *ad hoc*, no que diz respeito ao controle de acesso ao meio, criação e gerenciamento de rotas, mobilidade e segurança. As redes *ad hoc* são redes de propósito geral, nas quais não existe um nó central para onde as informações devem convergir, ou seja, todos os nós sensores funcionam como roteadores e a topologia é dinâmica. Embora tenham sido propostos diversos protocolos para as redes sem fio *ad hoc*, estes algoritmos não se adequam perfeitamente às necessidades e restrições das redes de sensores sem fio. Conforme encontrado em (AKYILDIZ et al., 2002), segue uma lista das principais diferenças entre RSSFs e redes *ad hoc*:

- O número de nós em uma RSSF pode ser muito maior do que a quantidade de nós em uma rede *ad hoc*;
- Os nós da RSSF são mais propensos a falhas, já que estão mais sujeitos a falta de energia e fatores ambientais;
- Os nós de uma RSSF são mais limitados em consumo de energia, capacidade de processamento computacional e memória em relação às redes *ad hoc* (BURATTI, 2011).

As principais características das redes de sensores sem fio são a escalabilidade (em relação ao número de nós da rede), auto-organização (a capacidade de recuperação de perdas de nós), eficiência energética, conectividade suficiente entre os nós (todos os nós da rede devem ser capazes de transmitir suas informações para o concentrador), baixa

complexidade, baixo custo e pequenos no tamanho. A definição de um protocolo que atenda a todas essas necessidades não é uma tarefa simples (BURATTI, 2011). A seguir serão abordados alguns fatores que devem ser levados em consideração durante o projeto de uma RSSF.

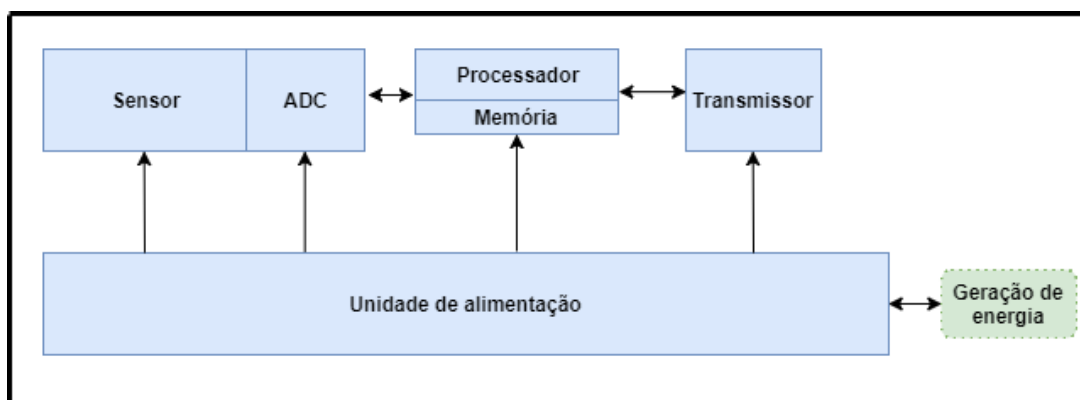
### 2.2.1.1 Tolerância a falhas

Os nós de uma rede de sensores sem fio podem apresentar algumas falhas, por exemplo, não possuir energia suficiente ou ser danificado fisicamente devido a alguma interferência do ambiente. As falhas de nós isolados da rede não devem afetar as tarefas executadas pelos demais elementos da RSSF. A tolerância a falhas é a capacidade de manter as funcionalidades da rede de sensores sem interrupções devido a falhas nos nós da rede (HOBLOS; STAROSWIECKI; AITOUACHE, 2000).

### 2.2.1.2 Restrições de Hardware

Os nós das RSSFs são constituídos geralmente de quatro componentes, uma unidade de sensoriamento, uma unidade de processamento, um transmissor e uma unidade de alimentação, como demonstrado na Figura 4. A unidade de sensoriamento obtém uma medida analógica do fenômeno mensurado, a qual é convertida para informação digital pelo conversor analógico-digital (ADC – *Analog-to-Digital Converter*). O processador tem a função de controlar a execução das tarefas de sensoriamento e os processos colaborativos entre os nós sensores (AKYILDIZ et al., 2002). A unidade de transmissão é responsável por conectar o nó à rede sem fio. A unidade de alimentação fornece energia para o nó sensor e demais componentes. Esta pode utilizar elementos de geração de energia, por exemplo, painéis solares.

Figura 4 – Diagrama de blocos do nó sensor de uma RSSF.



Fonte: Adaptado de (SILVA, 2011).

### 2.2.1.3 Custo de Produção

Tendo em vista que as aplicações podem requerer uma grande quantidade de nós sensores é de extrema importância que o custo de cada nó seja o mais baixo possível, a fim de minimizar o custo total da rede. Trata-se de uma restrição importante, pois caso o custo da rede seja mais elevado do que a implantação de sensores tradicionais, a rede não encontrará seu custo justificado (AKYILDIZ et al., 2002).

### 2.2.1.4 Manutenção da Topologia da Rede de Sensores

Os nós de uma rede podem ser dispostos de diversas maneiras. A implantação de vários nós sensores exige uma preocupação com a manutenção da topologia. Os problemas relacionados à topologia são divididos em três, pré-implantação, pós-implantação e adição de novos nós.

Na fase de pré-implantação é levado em consideração como os nós serão inicialmente implantados. Na fase de pós-implantação as preocupações são com as mudanças na topologia da rede causadas pela mobilidade dos nós ou falta de energia. Já a fase de adição de novos nós prevê a possibilidade de inserção e substituição de nós com defeito.

### 2.2.1.5 Consumo de Energia

A maior restrição das RSSFs está associada à eficiência energética, pois os nós são alimentados por baterias, as quais devem ser trocadas ou recarregadas periodicamente. Durante a transmissão, ocorre um pico de consumo de energia, drenando mais corrente da bateria quando apenas o processador e o sensor estão ativos. Protocolos de roteamento que exijam muitas transmissões de mensagens de controle devem ser evitados. Também não é desejado que o nó sensor permaneça durante muito tempo no estado de recepção, pois mesmo consumindo menos energia do que nas transmissões, este é um estado de alto consumo e a presença prolongada nesse estado deve ser evitada.

## 2.3 Vantagens das RSSFs

A maior vantagem das RSSFs está relacionada à robustez e escalabilidade. Um sistema de sensoriamento descentralizado é mais robusto a falhas do que um único nó sensor que concentra todas as comunicações, já que o sistema descentralizado possui redundâncias, que o torna mais tolerante a falhas (ZHAO; GUIBAS, 2004).

As RSSFs também são mais flexíveis que as redes de sensores que utilizam uma estrutura cabeada, já que é mais fácil inserir novos nós e não existe um custo elevado para remover um determinado nó ou replanejar a topologia da rede.

### 2.3.1 Eficiência Energética

Devido às características de atenuação dos sinais de radiofrequência (RF), uma rede sem fio que utiliza de múltiplos saltos proporciona uma economia de energia maior do que se fosse utilizada uma rede em que a comunicação é realizada em um único salto. O projeto deve buscar uma boa eficiência energética com um custo acessível. Latência e robustez também podem influenciar contra a decisão de ter um grande número de nós com um baixo alcance de transmissão (ZHAO; GUIBAS, 2004).

O gerenciamento eficiente de energia é um requisito fundamental para um projeto confiável de uma RSSF. As técnicas de economia de energia se concentram em dois subsistemas, o subsistema de rede (a gestão de energia é direcionada para as operações em cada nó único) e o subsistema de detecção (onde técnicas são utilizadas para reduzir a quantidade de amostras com alto consumo de energia). O tempo de vida útil de uma RSSF pode ser estendido através da aplicação de diferentes técnicas em conjunto. Por exemplo, os protocolos de eficiência energética propõem minimizar o consumo de energia durante as atividades da rede.

Para obter uma melhor eficiência energética de uma RSSF, deve-se analisar as características de dissipação de energia de um nó sensor da rede. A análise sistemática da potência de um nó sensor é extremamente importante para identificar os gargalos de alimentação no sistema, o que pode ser o alvo de otimização. A caracterização de cada unidade que compõem a arquitetura de um nó sensor é extremamente importante para a identificação dos possíveis pontos de desperdício ou consumo ineficiente de energia. É recomendado focar na otimização da rede visando redução do consumo de energia, na tentativa de prolongar o tempo de vida do nó sensor, e, conseqüentemente, prolongar o tempo de vida da rede.

## 2.4 Camadas das RSSFs

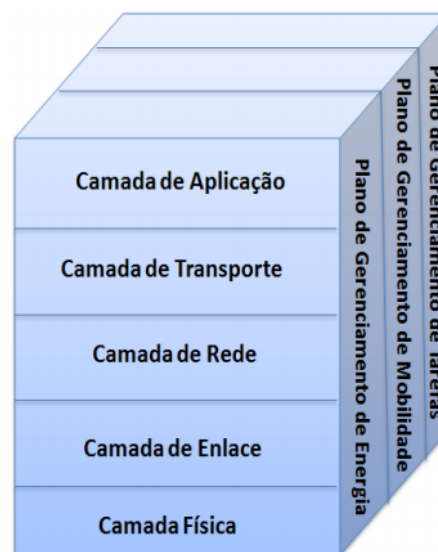
A pilha de protocolos de um nó sensor é demonstrada na Figura 5. Além das camadas podem ser observados os planos de gerenciamento. Os planos de gerenciamento buscam ajudar a coordenação dos serviços dos nós e a diminuição do consumo de energia (AKYILDIZ et al., 2002).

A camada física possui a responsabilidade de realizar a seleção da frequência de transmissão, a detecção de sinal, a geração de portadora de frequência e a modulação dos dados (AKYILDIZ et al., 2002). Para a frequência de transmissão, a faixa reservada para aplicações industriais, científicas e médicas (ISM – *Industrial, Scientific and Medical*) é composta por três faixas, 900 MHz, 2,4 GHz e 5,8 GHz. A faixa de 900 MHz não é utilizada no Brasil para fins comerciais, assim, pode ser utilizada na aplicação da rede de sensores sem fio. A faixa de 2,4 GHz é a mais utilizada, sendo compartilhada por diversas

aplicações como WiFi, Bluetooth e ZigBee (IEEE 802.15.4).

A camada de enlace é responsável principalmente pela detecção de quadros de dados, acesso ao meio e correção de erros. Ela permite a realização de comunicações ponto-a-ponto e ponto-a-multipontos na rede de sensores. Outra função importante da camada de enlace é o controle de erro. O controle de acesso ao meio (MAC) tem dois objetivos. O primeiro é a criação de uma infraestrutura de rede, ou seja, estabelecer links de comunicação para transferências de dados entre os nós e, o segundo, é a divisão dos recursos de forma justa e eficiente entre os nós.

Figura 5 – Pilha de protocolos das redes de sensores sem fio.



Fonte: Adaptado de (SILVA, 2011).

Nas redes multi-salto a camada de rede é responsável pela localização das rotas entre o nó fonte e o destino. As rotas devem levar em consideração a potência disponível nos nós ou a energia necessária para transmissão nos enlaces ao longo da rota. Uma rota entre nós fonte e destino pode ser escolhida por diversas abordagens, por exemplo, máximo de potência disponível nos nós, menor energia de transmissão ou menor número de saltos (AKYILDIZ et al., 2002).

Já a camada de transporte é necessária para que a RSSF possa ser acessada por outras redes externas, por exemplo, a Internet. Por fim, a camada de aplicação está diretamente relacionada à execução dos serviços da RSSF, no que diz respeito ao gerenciamento das tarefas de sensoriamento.

## 2.5 Classificação dos Protocolos para RSSFs

A topologia de uma rede móvel *ad hoc* pode mudar a qualquer instante. Assim, os nós sensores da rede precisam se conectar de forma dinâmica e arbitrária, comportando-se

como roteadores e assumindo o trabalho de descoberta e manutenção das rotas. Este trabalho de descoberta e manutenção de rotas é realizado pelo protocolo de roteamento, que tem como objetivo projetar a estratégia para transferir os dados de um nó para outro.

Existem três classificações para o envio de dados em redes *ad hoc*, são eles: *unicast*, *multicast* e *broadcast*. No envio do tipo *unicast* os dados são enviados de um ponto a outro, ou seja, ponto-a-ponto. Assim, este envio é adequado para redes cuja sua topologia seja alterada regularmente e que possua recursos limitados. No envio *multicast* os dados são enviados para múltiplos destinatários ao mesmo tempo; e no envio *broadcast*, a entrega ocorre para todos os pontos da rede. Os protocolos de roteamento *unicast* são classificados em proativos, reativos e híbridos (HEREK, 2011).

### 2.5.1 Protocolos Reativos

O roteamento reativo define-se por atuar sob demanda, ou seja, são criadas e mantidas as rotas somente quando requisitados por um nó. São protocolos que não preocupam-se em manter as informações de roteamento atualizadas constantemente. Quando o nó da rede sensor precisa enviar uma informação para outro nó e não possui as informações de roteamento necessárias, este nó envia uma solicitação de rota para todos os nós da rede, assim os nós que possuem a rota respondem ao nó de origem, que acaba por escolher a melhor rota entre as que recebeu através de seu algoritmo de seleção de rotas (HEREK, 2011).

O nó de origem normalmente precisa apenas encontrar o próximo salto, entretanto o processo de descoberta de rota acarreta em um atraso inicial, e os protocolos sob demanda devem ser tolerantes a esses atrasos. Um mecanismo utilizado para diminuir este atraso, é a utilização da cache do microcontrolador para armazenar as rotas já estabelecidas entre os nós fontes e destino, mas essas rotas possuem um prazo para expirar, pois num ambiente móvel as rotas ficarão inválidas depois de um determinado tempo (HEREK, 2011).

A vantagem dos protocolos reativos está no fato de que eles não necessitam ficar enviando muitas mensagens de controle pela rede para manter as rotas atualizadas, devido ao fato de trabalharem sob demanda. Assim, eles economizam os recursos energéticos e de processamento da rede e dos dispositivos. São exemplos de protocolos de roteamento reativos: AODV, DSR e TORA (HEREK, 2011).

### 2.5.2 Protocolos Proativos

Os protocolos proativos caracterizam-se por trabalhar para manter as informações de roteamento atualizadas. Esses protocolos são divididos em duas classes estratégicas: protocolos orientados a eventos e protocolos de atualizações regulares (HEREK, 2011).

Os protocolos orientados a eventos enviam os pacotes de atualização de roteamento

apenas quando a topologia da rede é alterada. A topologia de uma rede é alterada quando um novo nó aparece no raio de alcance ou quando o nó se move para fora do raio de alcance. Conforme ocorrem as alterações, o protocolo orientado a eventos envia a informação de alteração de acordo com sua estratégia de roteamento (HEREK, 2011).

Os protocolos de atualizações regulares enviam as informações de topologia em intervalos regulares independente dos eventos ocorridos na rede. A grande vantagem dos protocolos proativos está no fato de que as rotas podem ser usadas imediatamente, portanto não há atraso inicial para o envio da informação através rede de sensor. Entretanto, a mobilidade dos nós sensores pode ocasionar perdas de conectividade durante a descoberta de novas rotas válidas. Além disso, esses protocolos geram uma elevada carga de pacotes de controle. São exemplos de protocolos de roteamento proativos: DSDV, WRP e CGSR (HEREK, 2011).

### 2.5.3 Protocolos Híbridos

Os protocolos híbridos são os protocolos que utilizam ambas estratégias, reativa e proativa na manutenção de suas rotas. A característica pró-ativa ou reativa é utilizada de diferentes maneiras dependendo de cada protocolo híbrido.

## 2.6 Protocolo de Roteamento AODV

O *Ad Hoc On Demand Distance Vector Routing* (AODV) detém características como topologia dinâmica, que sofre mudanças constantes. É um protocolo reativo de forma a construir suas rotas só quando necessário. Assim, acontece um processo que utiliza medidas como envio de um grande número de pacotes para descobrir possíveis rotas, mostrando, dessa forma, uma grande característica do protocolo que é tentar aumentar a largura de banda disponível, procurando minimizar o fluxo de mensagens emitidas para atualizar suas rotas. Essa descoberta permite que cada nó móvel na rede atue como um roteador especializado e as rotas são obtidas quando necessárias, tornando, assim, uma rede autopartida. Cada nó na rede mantém uma tabela de roteamento com as entradas de informações de roteamento para os nós vizinho, onde só é guardado o próximo salto até o destino.

O AODV possui dois mecanismos de atualização da tabela de roteamento dos dispositivos, um de descoberta e outro de manutenção das rotas.

### 2.6.1 Descoberta de Rotas

O procedimento de descoberta de rota começa quando o nó fonte necessita enviar os dados para o nó destino e não existe rota disponível na tabela de roteamento. Essa



descoberta resume-se no envio de mensagens *Route Request* (RREQ) via *broadcast* para todos os nós vizinhos sobre a necessidade de uma rota. O pacote RREQ contém o identificador do *broadcast*, endereço da fonte e destino, número de sequência da fonte e do destino e contador de saltos (RIO., 2010).

Para cada nova mensagem RREQ enviada pelo nó fonte, o identificador do *broadcast* é incrementado. Então, se um nó intermediário receber um RREQ e posteriormente outro com o mesmo identificador e endereço da fonte, o último RREQ será descartado. Enquanto não for encontrada uma rota até o destino, os pacotes RREQ propagam-se para todos os nós próximos, sempre incrementando o contador de saltos e guardando as entradas temporárias em suas tabelas, que contêm as informações de endereço da fonte e do destino, identificador do *broadcast*, número de sequência da fonte e tempo limite da rota com o caminho de volta para fonte. A descoberta da rota continua até que o nó destino seja encontrado ou até um nó intermediário tenha a rota disponível até o destino. O número de sequência do destino mantém as informações das rotas atualizadas do nó fonte até o nó destino. O número de sequência da fonte é usado para verificar o quão recente são as informações da rota reversa (NETO, 2011).

No caso de um nó intermediário possuir a rota até o nó destino e receber um pacote RREQ, verifica-se se esta rota é a mais atual, através da comparação do número de sequência de destino do RREQ com o número de sequência de destino existente no nó. Se o número de sequência de destino do RREQ for maior, a rota do nó intermediário será então descartada, prevalecendo assim o que a que chegou primeiro, e o RREQ é retransmitido para os vizinhos. Caso contrário, um pacote *Route Reply* (RREP) via *unicast* com a rota reversa é enviado de volta para o nó fonte. O mesmo processo é realizado quando o próprio nó destino é encontrado. Esse pacote RREP possui as informações do endereço da fonte e do destino, número de sequência do destino, contador de saltos e o tempo de duração da rota. Para cada nó onde passou o pacote RREP, o nó armazena na sua tabela de roteamento a entrada com o próximo salto e a quantidade de saltos até o destino. Além de atualizar as informações do tempo de duração da rota e o último número da sequência até o destino. Apenas uma entrada é registrada para cada destino (NETO, 2011).

A cada nova rota armazenada na tabela de roteamento é associada a ela um tempo de vida, ou seja, se esta rota não for utilizada neste período de tempo ela será excluída da tabela.

## 2.6.2 Manutenção de Rotas

Devido à movimentação dos nós dentro da rede, pode ocorrer quebras de enlaces, isso faz com que rotas existentes nas tabelas de roteamento fiquem inválidas. Então, o protocolo AODV possui um mecanismo de manutenção de rotas, para verificar a validade dessas rotas (RIO., 2010).

Para detectar quebras de link, o protocolo envia mensagens *HELLO* para a rede afim de verificar se há falhas no link. As mensagens *HELLO* possuem o número de sequência e identificador da fonte. A informação de quebra de link é enviada através de um pacote identificador de erro *Route Error* (RERR) de volta para o nó de origem e todos os nós vizinhos que usam este enlace em alguma rota na sua tabela (NETO, 2011).

Se um nó não receber a mensagem do vizinho, é considerado que há uma quebra de link, assim, seus vizinhos propagam esta notificação de quebra de link até que nó fonte seja informado sobre a falha e possa enviar um novo RREQ. As rotas invalidas são removidas da tabela de roteamento conforme seus números de sequência. A mensagens *HELLO* são enviadas periodicamente para seus vizinhos via *broadcast*.

## 2.7 Protocolo de Roteamento DSR

O *Dynamic Source Routing* (DSR) é um protocolo de roteamento reativo, simples e eficiente, projetado especificamente para uso em redes *ad hoc* sem fio com nós móveis. O DSR permite que a rede seja completamente auto-organizada e autoconfigurada, sem a necessidade de qualquer infraestrutura ou administração de rede existente. Todos os aspectos do protocolo operam totalmente sob demanda, permitindo que a sobrecarga de pacotes de roteamento do DSR se ajuste automaticamente, e a apenas o necessário para reagir às mudanças nas rotas atualmente em uso. O protocolo permite rotas múltiplas para qualquer destino e que cada remetente selecione e controle as rotas usadas no roteamento de seus pacotes, por exemplo para uso em balanceamento de carga ou para maior robustez (NETO, 2011).

Os nós de rede cooperam para encaminhar pacotes um para o outro afim de permitir a comunicação através de múltiplos saltos entre nós que não estão diretamente dentro do alcance de transmissão. À medida que os nós da rede se movem ou saem da rede, todo o roteamento é determinado e mantido automaticamente pelo protocolo de roteamento. O protocolo DSR possui uma sobrecarga muito baixa, mas consegue reagir rapidamente às mudanças na rede. Também fornece um serviço altamente reativo para ajudar a garantir a entrega de pacotes de dados, apesar do movimento do nó ou outras mudanças nas condições da rede (NETO, 2011).

dentre as vantagens do protocolo DSR pode-se citar é o roteamento livre de *loop*, suporte para uso em redes contendo links unidirecionais, ou seja, pode armazenar múltiplas rotas para um só destino, e recuperação rápida quando rotas na rede mudam. O protocolo DSR foi projetado principalmente para redes móveis *ad hoc* de até duzentos nós, mesmo com taxas de mobilidade muito altas (RIO., 2010).

### 2.7.1 Descoberta de Rotas

Quando o nó fonte cria um novo pacote endereçado a algum nó de destino, o nó fonte coloca no cabeçalho do pacote uma rota de origem, dando a sequência de saltos que o pacote deve seguir no caminho até o destino. Normalmente, o remetente obterá uma rota de origem ao pesquisar na cache da tabela de roteamento as rotas previamente armazenadas. Caso nenhuma rota for encontrada em sua cache, iniciará o mecanismo de descoberta de rota para encontrar dinamicamente uma nova rota até o nó de destino (RIO., 2010).

Para iniciar a descoberta de rota, o nó fonte transmite um *Route Request* (RREQ) via *broadcast* como um único pacote de transmissão, que é recebido por todos os nós dentro da faixa de transmissão sem fio. Cada solicitação de rota carrega as informações do endereço da origem e do destino, identificador determinado pela origem e uma lista de endereços por onde o pacote tenha passado. Para identificar as requisições de rotas duplicadas, cada nó mantém na sua tabela o endereço de origem e o identificador do RREQ recebido (RIO., 2010).

Ao receber um pacote RREQ, o nó intermediário verifica se ele é o destinatário. Se sim, ele envia um pacote *Route Reply* (RREP) para o nó fonte, incluindo a lista da sequência de todos os nós aglomerados pelo RREQ até o destino. Se não for, o nó sensor confere se a requisição de rota já tenha sido recebida anteriormente, caso contrário, o pacote é descartado. Caso não tenha recebido anteriormente, o nó verifica se possui uma rota válida em sua cache até o destino. Caso possua, o pacote RREP é enviado até o nó fonte com o caminho de todos os nós intermediários acumulados pelo RREQ. O RREP é enviado via *unicast* seguindo a rota reversa do RREQ (NETO, 2011).

O destinatário pode receber várias requisições de rotas com o mesmo identificador do RREQ e o endereço de origem. Assim, o nó destino responderá todos os RREQ enviando os pacotes RREP ao nó fonte, assegurando rotas alternativas da fonte até o destino (RIO., 2010).

### 2.7.2 Manutenção de Rotas

O nó que transmite o pacote é responsável por confirmar se os dados foram entregues ao nó destinatário, ou seja, cada nó fica responsável por verificar o recebimento de pacotes do vizinho. Enquanto uma rota está em uso, o procedimento de manutenção de rota monitora a operação da rota e informa ao remetente qualquer erro de roteamento, através do uso de um pacote *Route Error* (RERR) e confirmações de recebimento de pacotes. Ao receber o pacote de erro, o nó remove de sua tabela de roteamento todas as rotas que possuem o endereço do link quebrado (NETO, 2011).

A vantagem do protocolo DSR está na não utilização de mensagens *HELLO* para

o reconhecimento dos nós vizinhos, como é utilizado no protocolo AODV, assim, aumenta a largura de banda disponível e diminui o consumo de energia dos nós. Outra vantagem do DSR é que os nós observam os pacotes que estão trafegando pela rede e que possam reunir os caminhos que estão contidos nos pacotes sem que estes participem do processo de descoberta de rota (RIO., 2010).

## 2.8 Protocolo de Roteamento DSDV

O *Destination Sequenced Distance Vector* (DSDV) é um protocolo proativo, baseado no mecanismo de roteamento Bellman-Ford. O DSDV é o resultado de uma das primeiras tentativas de adaptação de um mecanismo de roteamento para a aplicação em redes móveis *ad hoc*. A principal melhoria aplicada ao algoritmo inicial é a capacidade de evitar *loops* na rede (RIO., 2010).

Todos os nós da rede possuem uma tabela de roteamento com o endereço do nó destino. Essas tabelas de roteamento são atualizadas periodicamente para que possam ter consistência dos dados armazenados. As mensagens de atualização também são transmitidas quando se percebe uma alteração no estado dos enlaces.

### 2.8.1 Descoberta de Rotas

Cada nó na rede mantém uma tabela de roteamento com todos os destinos possíveis, o número de saltos até o destino e um número sequencial atribuído pelo nó destino. Este número sequencial é utilizado para distinguir novas rotas a partir de rotas antigas, e também evitam *loops* na rede. Em uma atualização, a rota com o número sequencial mais recente será utilizada, enquanto a mais antiga é descartada. Quando uma rota com o mesmo número sequencial é recebida, a rota com o menor número de saltos será a rota preferencial, enquanto a outra rota é eliminada ou armazenada como não preferencial (HEREK, 2011).

### 2.8.2 Manutenção de Rotas

A estratégia de manutenção de rotas proativa, faz a tabela de roteamento do protocolo DSDV ser atualizada constantemente. Como solução para evitar congestionamento no meio foram implementados dois tipos de atualizações: completa e incremental. Na atualização completa, também chamada de *full dump*, todas as informações de roteamento disponíveis são transmitidas, e podem exigir o envio de múltiplos *network protocol data units* (NPDUs). Quando existem poucas mudanças de topologia, as atualizações completas são menos frequentes. Por outro lado, as atualizações incrementais utilizam necessariamente, por determinação do projeto do protocolo, apenas uma NPDU. As atua-

lizações incrementais são utilizadas para trafegar as mudanças ocorridas depois da última transmissão completa e ajudam a diminuir o tráfego gerado ([HEREK, 2011](#)).

## 2.9 Comentários

Nesta seção foi apresentada a fundamentação teórica sobre redes sensores sem fio necessária para entendimento deste trabalho, como as definições de termos utilizados, as vantagens, as camadas e os protocolos de roteamento. Foram utilizados os protocolos AODV, DSDV e DSR, pois são os mais empregados em redes sensores sem fio de baixo consumo de energia e são os três protocolos que o simulador NS-2 dispõe para simulação.

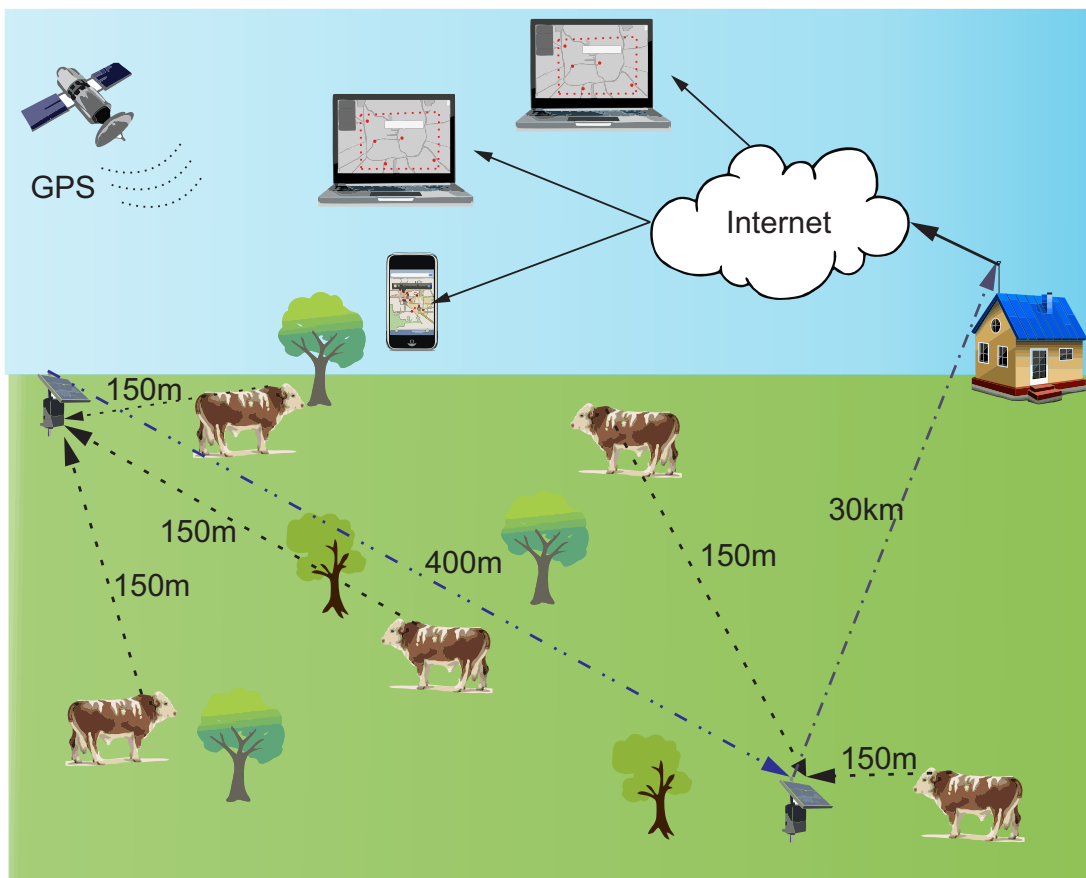
## 3 Metodologia

### 3.1 Arquitetura da RSSF

A arquitetura proposta para a RSSF foi uma *ad hoc*, na qual não existe um nó central para onde todas as informações devem convergir. Na rede haverá apenas um concentrador, onde este é responsável pelo roteamento dos dados recebidos (localização) dos nós sensores para o *gateway*. O *gateway* enviará as localizações dos sensores para a Internet, onde os usuários possam ter acesso as informações de localização de cada nó sensor.

Considere a Figura 6, onde cada boi representa um nó sensor da rede, cada “painel” representa um concentrador e a casa apresenta o *gateway* da rede de sensores sem fio.

Figura 6 – Exemplo da Rede Sensor Sem Fio na Prática.



Fonte: Autor.

As informações sobre localização são obtidas a partir de módulos de recepção de sinais GPS em cada nó sensor, que processa essa informação e, posteriormente, envia a informação para o concentrador ou senão, reencaminha a informação para o nó sensor

mais próximo do centralizador ou ainda reencaminha novamente a informação para o nó mais próximo dele até chegar no centralizador.

Após, as informações chegarem no centralizador, os mesmos irão rotear as informações para o *gateway* (casa). Posteriormente estas são disponibilizadas na Internet para que os usuários (pecuaristas) possam ter a localização exata dos seus animais.

## 3.2 Componentes Utilizados

Inicialmente, o objetivo do trabalho era implementar fisicamente a rede de sensores sem fio com os componentes existentes no laboratório. Assim, as características e especificações dos componentes foram empregados no ambiente de simulação.

### 3.2.1 Processador

O Arduino Uno é uma placa de microcontrolador baseado no ATmega328. Ele tem 14 pinos de entrada/saída digitais (dos quais 6 podem ser usados como saídas PWM), 6 entradas analógicas, um cristal oscilador de 16MHz, uma conexão USB, uma entrada de alimentação, uma conexão ICSP e um botão de reset ([ARDUINO, 2016](#)).

O arduino Uno pode ser alimentado por conexão USB ou pelo conector de alimentação de 2,1mm, ou ainda, por uma fonte de alimentação externa. A fonte de alimentação é selecionada automaticamente. Alimentação externa (não USB ou conector de 2,1mm) pode ser tanto de um adaptador CA para CC ou uma bateria de 9V. Os cabos vindos da bateria podem ser inseridos diretamente nos pinos Gnd e Vin ([ARDUINO, 2016](#)).

Esta placa pode funcionar com uma fonte de alimentação externa de 6 a 20 volts. No entanto se a alimentação for inferior a 7V, o pino de 5V pode fornecer menos de cinco volts e a placa pode se mostrar instável. E se a alimentação for maior do que 12V o regulador de tensão pode superaquecer e danificar a placa. A faixa recomendada é de 7 a 12 volts ([ARDUINO, 2016](#)).

Os pinos de alimentação são os seguintes:

- **VIN:** Entrada de alimentação para a placa Arduino quando emprega-se uma fonte de alimentação externa. Pode-se fornecer alimentação através deste, ou utilizar a tensão do pino (5V) quando se estiver utilizando a conexão USB ou o conector de alimentação como fonte para a placa.
- **5V:** Fonte de alimentação regulada usada para o microcontrolador e para outros componentes externos, por exemplo, *shields* para o arduino.
- **3V3:** Fonte de 3,3V gerada pelo regulador de tensão embarcado. A corrente máxima suportada pela porta é de 50mA.

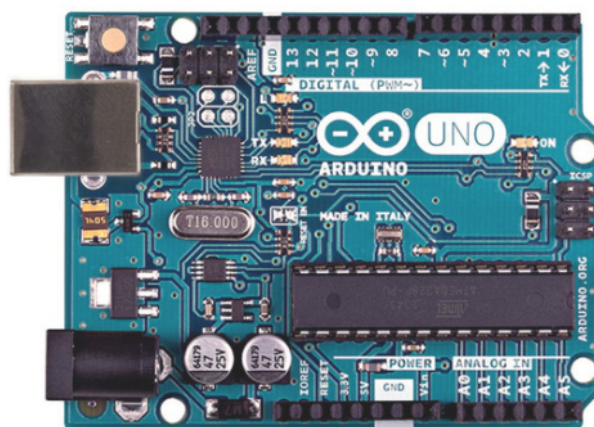
- **GND:** Pino de terra.

O ATmega328 têm 32KB de memória *flash* (dos quais 0,5KB são utilizados pelo bootloader) e também tem 2KB de SRAM e 1KB de EEPROM (ARDUINO, 2016).

O Arduino Uno possui uma série de facilidades para se comunicar com um computador, outro Arduino, ou outros microcontroladores. O ATmega328 fornece comunicação serial UART TTL (5V) que está disponível nos pinos digitais 0 (RX) e 1 (TX). Um ATmega8U2 na placa canaliza esta comunicação para o USB e aparece como uma porta virtual para o software no computador. O *firmware* do ATmega8U2 utiliza os *drivers* padrões USB COM e nenhum outro *driver* externo é necessário. O software do Arduino inclui um monitor serial que permite o envio e recebimento de dados textuais. O ATmega328 também suporta comunicação I2C e SPI (ARDUINO, 2016).

O Arduino Uno pode ser programado com o software Arduino®. O ATmega328 no Arduino Uno vem pré-gravado com um bootloader que permite a você enviar código novo para ele sem a utilização de um programador de hardware externo (ARDUINO, 2016). Um exemplo de Arduino Uno é demonstrado na Figura 7.

Figura 7 – Arduino Uno.



Fonte: Adaptado de (ARDUINO, 2016)

Na Tabela 1 são apresentadas as especificações técnicas do Arduino Uno.

### 3.2.2 Transceptor

O nRF905 é um transceptor (transmissor e receptor) de rádio de único chip para a banda ISM (*Industrial Scientific and Medical*) de 433/868/915 MHz. O transceptor consiste em um sintetizador de frequência integrado, um receptor com demodulador, um amplificador de potência, um oscilador de cristal e um modulador. O transceptor é demonstrado na Figura 8 (SEMICONDUCTOR, 2008).

O nRF905 possui as duas primeiras camadas de rede do modelo OSI (*Open System Interconnection*) implementadas utilizando a tecnologia ShockBurst™ que se encarrega



Tabela 1 – Especificações Técnicas.

Microcontrolador	ATmega328
Tensão de Entrada (recomendada)	7 - 12 V
Tensão Operacional	5 V
Pinos de I/O Digitais	14
Pinos PWM	6
UART	1
SPI	1
I2C	1
Pinos de Entrada Analógica	6
Interrupções Externas	2
Corrente DC por Pino de I/O	40 mA
Memória Flash	32 KB
SRAM	2 KB
EEPROM	1 KB
Frequência de Clock	16 MHz

Fonte: Adaptado de ([ARDUINO, 2016](#))

da gestão dos pacotes, verificação de erros e retransmissões. Esta tecnologia opcional denominada Enhanced Shockburst está embutida em seu *firmware*. Deste modo, consegue-se obter a camada de nível 2 do modelo OSI totalmente gerida por *hardware* no módulo RF. Adicionalmente, existe uma porta de interrupção que permite notificar o microcontrolador quando ocorre uma mudança no estado do módulo de RF, que juntamente com a utilização da tecnologia Enhanced Shockburst, permite libertar o microcontrolador de diversas tarefas relativas à gestão do módulo de RF ([SEMICONDUCTOR, 2008](#)).

Uma vez que o módulo de RF implementa apenas o básico indispensável numa rede (primeiras duas camadas, física e de enlace) oferece uma grande flexibilidade no desenvolvimento de novas camadas sobre as já implementadas, embora estas tenham que ficar a cargo do microcontrolador ([SEMICONDUCTOR, 2008](#)).

O nRF905 não dispõe de quaisquer mecanismos de segurança ou camada de rede implementados nativamente. A camada de rede deve ser desenvolvida e implementada no Arduíno, ou microcontrolador empregado, que pode ser visto como uma vantagem ou desvantagem. O chip é disponibilizado em uma PCB juntamente com uma antena externa, que apresenta algumas limitações quanto à sua miniaturização, mas ao mesmo tempo cobre distâncias superiores. O consumo de corrente é baixo, 9 mA na transmissão, com uma potência de saída de -10dBm, e no modo de recepção 12,5 mA ([SEMICONDUCTOR, 2008](#)). Na Tabela 2 são apresentadas as especificações técnicas do nRF905.

Tabela 2 – Especificações Técnicas do nRF 905.

Parâmetro	Valor	Unidade
Tensão de Alimentação	3,3	V
Tensão de Alimentação Máxima	3,6	V
Tensão de Alimentação Mínima	1,9	V
Potência Máxima de Transmissão	10	dBm
Potência Mínima de Transmissão	-10	dBm
Taxa de Transmissão	50	kbps
Corrente de Alimentação na Transmissão de 10 dBm	30	mA
Corrente de Alimentação na Transmissão de -10 dBm	9	mA
Corrente de Alimentação no Modo de Recepção	12,5	mA
Faixa de Temperatura	-40 até +85	°C
Sensibilidade Típica	-100	dBm

Fonte: Adaptado de ([SEMICONDUCTOR, 2008](#)).

Figura 8 – nRF 905.



Fonte: Adaptado de ([SEMICONDUCTOR, 2008](#)).

### 3.2.3 Módulo GPS

O módulo GPS utilizado na rede foi um módulo GPS padrão com antena de 25\*25\*4 mm, de alta sensibilidade. Possui interface opcional UART/TTL e RS232, 56 canais de recepção e utiliza o sistema de coordenadas WGS-84. Na Figura 9 é demonstrado o módulo GPS ([ELECTRONICS, 2012](#)). Na Tabela 3 são apresentadas as especificações técnicas do módulo GPS. São fornecidas na Tabela 4 as definições dos pinos do módulo GPS.

## 3.3 Comentários

Nesta seção foram apresentados os componentes eletrônicos que seriam empregados na implementação da rede física. Devido ao baixo número de componentes dispostos no

Tabela 3 – Especificações do Módulo GPS.

Parâmetro	Valor	Unidade
Tensão de Alimentação	5	V
Tensão de Alimentação Máxima	5,5	V
Tensão de Alimentação Mínima	3,3	V
Sistema de Coordenadas	WGS-84	-
Canais de Recepção	56	-
Corrente de Alimentação	30	mA
Corrente de Alimentação Máxima	45	mA
Corrente de Alimentação Mínima	25	mA
Faixa de Temperatura	-40 até +85	°C
Sensibilidade Típica	-160	dBm

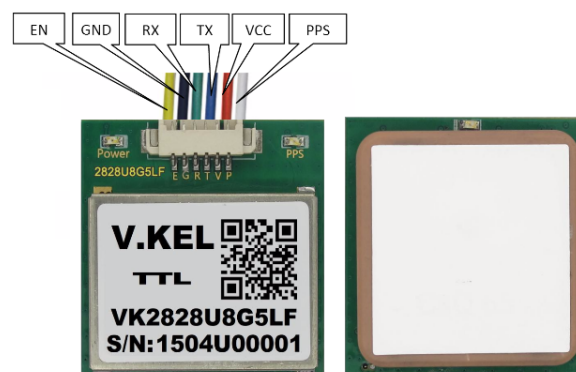
Fonte: Adaptado de (ELECTRONICS, 2012)

Tabela 4 – Definição dos pinos do módulo. Fonte: Adaptado de (ELECTRONICS, 2012)

Nome	Descrição
PPS	Pulsos de Tempo de Saída (pulso por segundo)
VCC	Tensão de Alimentação
TX	Interface UART/TTL e RS232_TXD é opcional
RX	Interface UART/TTL e RS232_RXD é opcional
GND	Conectar ao terra
EN	Habilitação de Potência: nível alto opera, nível baixo não opera

Fonte: Adaptado de (ELECTRONICS, 2012)

Figura 9 – Módulo GPS. Adaptado de (ELECTRONICS, 2012).



Fonte: Adaptado de (ELECTRONICS, 2012)

laboratório, optou-se por projetar a rede em um ambiente de simulação. Com isto, é possível analisar diferentes cenários e protocolos de comunicação de forma mais rápida. Estes mesmos componentes foram utilizados nas configurações de simulação do software NS-2. Destaca-se que os nós sensores a serem empregados foram desenvolvidos em laboratório e algumas medidas de consumo de energia foram efetuadas. Entretanto, estes dados não

são aqui apresentados pois o trabalho está delimitando apenas nos dados de simulação.

## 4 Estudo de Caso

### 4.1 Network Simulator 2

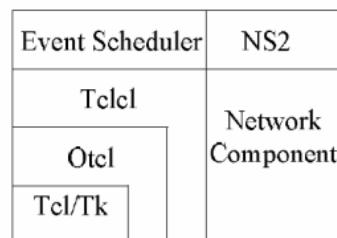
O NS-2 (*Network Simulator – versão 2*) é um software desenvolvido por pesquisadores da Universidade de Berkeley para simulações de redes de computadores, tem sua distribuição gratuita e seu código fonte aberto (VERONA, 2010). Graças a isso, teve grande avanço no seu desenvolvimento e larga utilização em pesquisas acadêmicas, onde tornou-se uma grande ferramenta colaborativa.

A essência básica do NS-2 é ser um simulador de eventos discretos, ou seja, é uma ferramenta dinâmica, a qual sofre alterações à medida que os eventos vão ocorrendo, como o envio de um determinado dado pela rede. Possui um relógio virtual, o qual orienta os eventos discretos que ocorrem durante a simulação e possui diversos módulos que podem simular inúmeros aspectos de uma determinada tecnologia de rede (SANTOS, 2011). Tornou-se uma das principais ferramentas de simulações de redes, principalmente no meio acadêmico, auxiliando o ensino e a pesquisa na área de redes de computadores.

Basicamente o simulador NS-2 necessita dos seguintes componentes para o seu funcionamento (MUCHA, 2013):

- **Tcl** – *Tool Command Language* e **Tk** – *ToolKit graphical user interface*;
- **OTcl** – Tcl orientada a objetos;
- **TclCl** – *Tool Command Language with Classes*.

Figura 10 – Componentes de software do simulador NS-2. .



Fonte: Adaptado de (MUCHA, 2013)

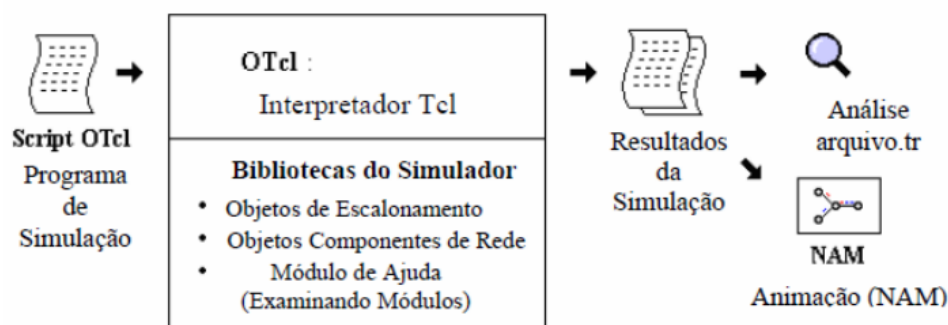
No modelo utilizado pelo NS-2, os componentes de rede (*networks components*) são considerados o núcleo do simulador, sendo escritos na linguagem C++ e OTcl. OTcl é uma extensão desenvolvida na linguagem C que proporciona a realização da programação orientada a objeto na linguagem Tcl (MUCHA, 2013).

O NS-2 necessita de um *script* escrito na linguagem OTcl para a realização de uma simulação, onde as configurações e parâmetros são definidos. Informações como o protocolo de roteamento, número de dispositivos de rede e tempo de simulação, estão presentes nesse *script*. Dessa forma, o NS-2 pode ser considerado como um interpretador de linguagem OTcl que possui um simulador de eventos discretos e bibliotecas com componentes escritos em linguagem C++ para simulação da rede (MUCHA, 2013).

O Tk é um componente de software utilizado no desenvolvimento de uma interface gráfica, a qual permite a visualização dos eventos que ocorreram durante a simulação. Ainda, o componente Tclcl proporciona uma interface entre o NS-2 e o OTcl, realizando a ligação entre objetos e as variáveis das duas linguagens (MUCHA, 2013).

Dessa forma, através do *script* OTcl, o NS-2 irá interpretar suas bibliotecas, obedecendo uma hierarquia de classes, cujo resultado será a criação de dois arquivos, um que demonstra o comportamento e o outro os resultados obtidos durante a simulação.

Figura 11 – Ilustração do processo de criação e resultados da simulação.



Fonte: Adaptado de (MUCHA, 2013).

Após a realização da simulação, o NS-2 gera arquivos de saída baseados em texto, possuindo informações detalhadas sobre a simulação efetuada. Uma dessas ferramentas que permite a visualização gráfica é o NAM (*Network Animator*). O NAM possibilita visualizar os fluxos de dados que trafegam nos enlaces e a ocupação das filas, entre outros fenômenos que ocorreram na rede, possui uma interface de trabalho simplificada, tornando-se amigável ao usuário, podendo apresentar graficamente informações presentes nas simulações (VERONA, 2010).

O NS-2 foi escolhido como ferramenta de simulação deste trabalho porque além das características já descritas, é um software livre bastante difundido no meio acadêmico e com ampla documentação disponível.

## 4.2 Ambiente e Cenários de Simulação

As simulações foram realizadas através do NS-2, na versão 2.35. As simulações foram realizadas em um computador com processador *Intel® Core i3* de 2,3GHz de sexta

geração com uma *Intel® HD Graphics 520* com memória RAM DDR3 de 4GB e 1TB de HD. A Tabela 5 apresenta as configurações dos parâmetros de hardware utilizado nas simulações.

Tabela 5 – Configurações dos Parâmetros de Hardware.

<b>Configurações do Hardware</b>	
Canal de Comunicação	Channel/WirelessChannel
Modelo de Propagação	Propagation/TwoRayGround
Interface de Rede	Phy/WirelessPhy
Camada MAC	802.11
Camada Física	LL
Tipo da Fila	Queue/DropTail/PriQueue
Tamanho da Fila	50
Tipo da Antena	Antenna/OmniAntenna
Frequência	433 MHz

A Tabela 6 apresenta as configurações em relação aos parâmetros de topologia e tráfego de dados da rede.

Tabela 6 – Configurações da Topologia e Tráfego de Dados da Rede.

<b>Parâmetros</b>	<b>Valores</b>
Protocolo de Roteamento	AODV/DSDV/DSR
Topologia da Rede	Aleatória
Tamanho da Rede	500 metros X 500 metros (x X y)
Alcance do Sinal	100 metros
Número de Nós	30
Agente de Tráfego	TCP/FTP
Tipo de Dados	TCP
Velocidade dos Nós Móveis	3 m/s ou 10,8 km/h
Tempo de Simulação	1800 segundos ou 30 minutos

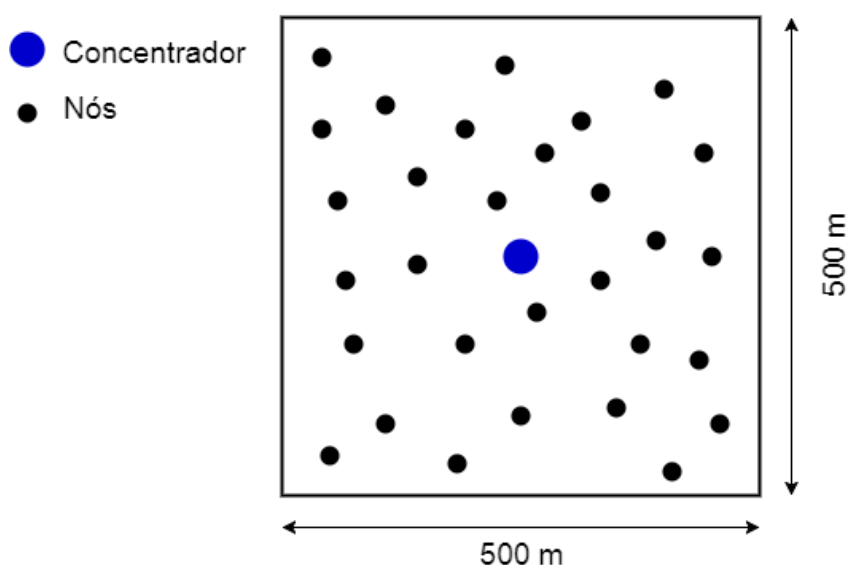
O tempo de simulação utilizado para todas as simulações foi de 30 minutos, pois o resultado de simulação no NS-2 é proporcional ao tempo e aos eventos ocorridos na simulação. O tempo máximo de simulação que foi possível abrir o arquivo *trace* dos resultados foi de 30 minutos.

As simulações foram divididas por protocolos de roteamento e por intervalos de tempo para as transmissões dos dados. Os protocolos de roteamento utilizados nas simulações foram AODV, DSR e DSDV. Através dos protocolos foram comparados os resultados de *overhead*, *throughput*, PDR e atraso médio. Os intervalos de tempo para as transmissões foram de 5 e 10 minutos entre cada transmissão. Por meio do intervalo de tempo foi possível analisar o consumo dos nós e o consumo médio da rede, além dos níveis de energia restante nos nós sensores.

O cenário de simulação criado no NS-2 foi de uma pastagem de 25 hectares, ou seja, 500m x 500m, onde se encontram 30 bovinos no pasto. O número de 30 bovinos refere-se ao número de hectares dividido pelo número de bovinos que coloca-se em um hectare. O número de bovinos por hectare é de 0,85, dados utilizados por pecuaristas da região. O valor exato foi de 29,41 bovinos em 25 hectares, então colocou-se 30 bovinos.

A Figura 12 apresenta o cenário físico da simulação, com o concentrador no centro da rede e os nós espalhados em volta. Esse cenário não é alterado, somente o cenário dos intervalos de tempo de transmissão dos dados. A Figura 13 apresenta o cenário da RSSF no NS-2, onde os nós se movimentam de maneira aleatória. O nó 0 (zero) é representado pelo concentrador no meio do cenário e os demais são os nós sensores.

Figura 12 – Cenário Físico da RSSF.



Fonte: Autor.

### 4.3 Configuração do Consumo de Energia

O NS-2 possui um modelo de energia, `energyModel`, responsável por definir o nível de energia de cada nó da rede durante a simulação. Através deste modelo, pode-se especificar a quantidade de energia que deve ser consumida em determinada tarefa (BERKELEY et al., 2011). A Tabela 7 demonstra o que pode ser configurável nesse modelo.

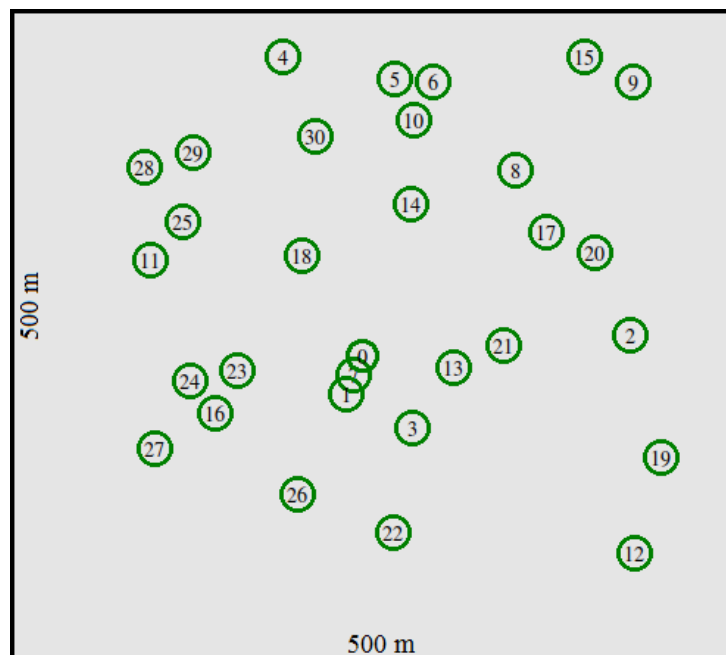
Após simulação do Modelo de Energia, a saída retorna um arquivo no seguinte formato de rastreamento,

```
[energy 0.000 ei 0.000 es 0.000 et 0.000 er 0.000]
```

onde,



Figura 13 – Cenário Físico da RSSF.



Fonte: Autor.

Tabela 7 – Parâmetros configuráveis do Modelo de Energia.

Atributo	Significado	Unidade
energyModel	Tipo do Modelo de Energia	EnergyModel
initialEnergy	Energia Inicial do Nó	Energia em Joules
rxPower	Potência para Receber um Pacote	Potência em Watts
txPower	Potência para Transmitir um Pacote	Potência em Watts
idlePower	Potência Consumida no Modo Ocioso	Potência em Watts
sleepPower	Potência Consumida no Modo Dormir	Potência em Watts
sensePower	Potência Consumida pelo Sensor	Potência em Watts

energy: Energia total restante;  
 ei: Energia consumida no estado ocioso;  
 es: Energia consumida no estado dormir;  
 et: Energia consumida na transmissão de pacotes;  
 er: Energia consumida na recepção de pacotes;

Todas as simulações utilizaram os valores apresentados na Tabela 8 para o Modelo de Energia.

O valor inicial de energia é referente a uma bateria de 10 Joules. Os valores das potências de transmissão e recepção são referentes a tensão de operação do transmissor pela corrente máxima de alimentação em cada modo, os valores de tensão e corrente de alimentação foram retirados da Tabela 2. A potência do sensor é o resultado da tensão de operação pela corrente de alimentação máxima, conforme apresentado na Tabela 3. A potência do processador está englobada na potencia do sensor, devido ao tempo de atualização do sensor ser na casa dos micros segundos e o valor de potência de ambos ser

Tabela 8 – Valores utilizados nas simulações no Modelo de Energia.

Estado	Valor	Unidade
initialEnergy	10	J
txPower	0,099	W
rxPower	0,042	W
idlePower	-	W
sleepPower	0,00017	W
sensePower	0,25	W

o mesmo.

O valor da potência consumida no modo *sleep* foi obtida a partir de medidas em laboratório, pois os *datasheets* dos componentes do nó sensor não apresentavam essa informação.

## 4.4 Métricas de Desempenho

A execução da simulação no NS-2 gera um arquivo *trace* (.tr) onde estão registrados todos os eventos ocorridos durante o processo de simulação. O arquivo *trace* é um arquivo texto, que descreve os eventos ocorridos na simulação (uma entrada em fila, uma saída de fila, um descarte de pacote ou um recebimento de pacote), o tempo de ocorrência, a identificação do pacote, o protocolo de transmissão, a identificação única do pacote e os nós de partida inicial e destino final do pacote (BERKELEY et al., 2011).

Com o arquivo *trace* desenvolveu-se um *script* em linguagem *awk* para extrair algumas métricas de desempenho para avaliação dos protocolos de roteamento. Os *scripts awk* desenvolvidos são para calcular o *overhead*, *throughput*, PDR, atraso médio fim-a-fim e o consumo médio de energia da rede. As métricas de desempenho utilizadas são descritas a seguir:

- *Overhead* ou Carga de Roteamento Normalizada: é a relação entre a quantidade de pacotes de roteamento enviados por pacotes de dados entregues. Essa métrica visa identificar em cada protocolo quantos pacotes de roteamento são necessários para que um pacote de dados seja entregue (SHARMA; GUPTA, 2012).
- *Throughput* ou Vazão: É a quantidade de dados transferidos em um período de tempo, ou seja, é a quantidade total de bits recebidos por segundo. Para calcular o *throughput* é medido o número de pacotes recebidos com sucesso. Em seguida, é computado o número de bits recebidos nesse nó. Esse número é então dividido pelo

tempo total de simulação (SHARMA; GUPTA, 2012).

$$\text{Throughput} = \frac{\text{Total de Bits Recebidos}}{\text{Tempo Total de Simulação}} \quad (4.1)$$

- *Packet Delivery Ratio (PDR)* ou Relação de Entrega de Pacotes: É a razão entre o número de pacotes recebidos pelo destino e os pacotes gerados pela fonte. A taxa de entrega indica a porcentagem de pacotes de informação transmitidos que são recebidos com sucesso (SHARMA; GUPTA, 2012).

$$\text{Delivery Ratio} = \frac{\text{Número de Pacotes Recebidos}}{\text{Número de Pacotes Transmitidos}} * 100\% \quad (4.2)$$

- *Atraso Médio Fim-a-Fim*: É o tempo decorrido desde a geração do pacote na fonte até o seu recebimento no destino. Inclui também o atraso causado pelo processo de descoberta de rota e a fila na transmissão de pacotes de dados. Apenas os pacotes de dados que foram entregues com sucesso ao destino são contabilizados (SHARMA; GUPTA, 2012).
- *Consumo Médio de Energia*: É a média da energia consumida por todos os nós juntos, ou seja, o consumo médio de energia da rede.
- *Porcentagem de Energia Consumida*: É a razão da energia consumida pela energia inicial.

$$\text{Porcentagem de Energia Consumida} = \frac{\text{Energia Consumida}}{\text{Energia Inicial}} * 100\% \quad (4.3)$$

## 4.5 Análise dos Resultados

Os resultados estão divididos em subseções para facilitar a análise de cada cenário. O primeiro cenário é a transmissão dos dados a cada 5 minutos, e o segundo cenário é a transmissão dos dados a cada 10 minutos. Serão analisados os resultados de *overhead*, *throughput*, PDR, atraso médio e consumo de energia para os três protocolos de roteamentos (AODV, DSR e DSDV) para cada cenário simulado.

### 4.5.1 Intervalo de Transmissão de 5 Minutos

A Tabela 9 demonstra os resultados obtidos no primeiro cenário de simulação, onde os dados foram transmitidos em intervalos de 5 minutos. Analisando os resultados da

Tabela 9 pode-se ver que o protocolo DSR apresenta uma melhor PDR em relação aos protocolos AODV e DSDV, isto é, 99,75% das transmissões foram realizadas com sucesso. Além disso, apresentou menor *overhead* e menor consumo médio da bateria, isto é, a rede com o protocolo DSR iria durar mais tempo que os outros protocolos.

O protocolo DSDV apresentou maior consumo médio da bateria, atraso médio fim-a-fim e *overhead* em relação aos outros protocolos, isto é, a rede iria parar de operar antes, além dos dados demorarem mais para ser entregues ao concentrador. Entretanto apresentou uma maior vazão (*throughput*) que os outros protocolos. O protocolo AODV apresentou menor atraso médio fim-a-fim.

Tabela 9 – Resultados do Primeiro Cenário

	Protocolo de Roteamento		
	AODV	DSR	DSDV
<b>Pacotes Enviados</b>	19603	19366	28053
<b>Pacotes Recebidos</b>	19454	19319	22397
<b>PDR (%)</b>	99,23	99,75	79,83
<b>Throughput (kbps)</b>	51,97	51,46	59,87
<b>Overhead</b>	0,571	0,529	0,598
<b>Atraso Médio Fim-a-Fim (ms)</b>	156,453	198,963	225,86
<b>Consumo Médio da Bateria (%)</b>	62,75	55,44	66,43

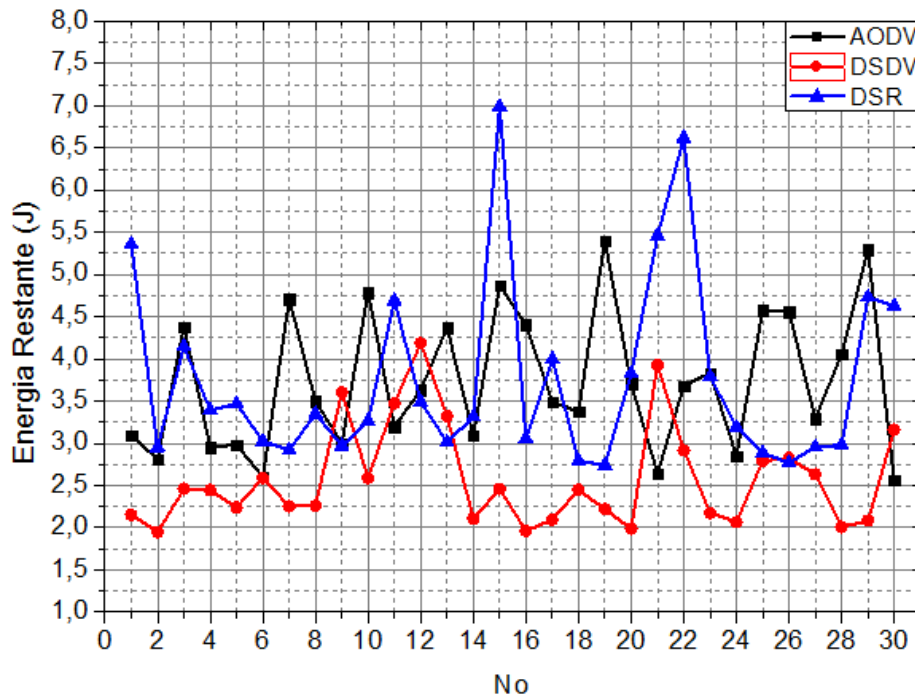
Considerando os resultados de consumo médio da bateria dos três protocolos acima, foi utilizado como exemplo uma bateria comercial para estipulado o tempo médio que a rede iria durar (resistir). Essa bateria é recarregável e tem capacidade de 175mAh com tensão nominal de 8,4V volts. Isto dá uma capacidade em Joules de 5292J. A capacidade da bateria depende da característica da descarga ([ENERGIZER](#), ).

Fazendo uma estimativa para a bateria com 10J, a rede com o protocolo AODV iria durar 0,8 horas. Já a bateria de 5292J a rede vai durar 529 vezes mais, então a rede vai durar 852,48 horas ou 17,63 dias. Com os protocolos DSR e DSDV e bateria de 5292J, a rede iria durar 19,83 e 16,53 dias, respectivamente.

A Figura 14 demonstra o gráfico da energia restante em cada nó da rede para transmissão a cada 5 minutos entre os três protocolos. Como pode-se analisar o protocolo DSR apresentou a maior energia restante para seus nós. Essa diferença entre as energias restantes dos nós entre os protocolos é justificada pelo *overhead* que cada protocolo apresentou, ou seja, quanto maior o *overhead*, mais tempo ligado fica o módulo para encontrar/atualizar as rotas de transmissão e mais energia o nó consome.

As Figuras 15, 16 e 17 apresentam a porcentagem de energia consumida por cada nó da rede para seus respectivos protocolos de roteamento. As diferenças de energia consumida entre os nós com o mesmo protocolo de roteamento ocorre porque os nós se movimentam de maneira aleatória, assim, fazendo com que alguns nós estejam mais

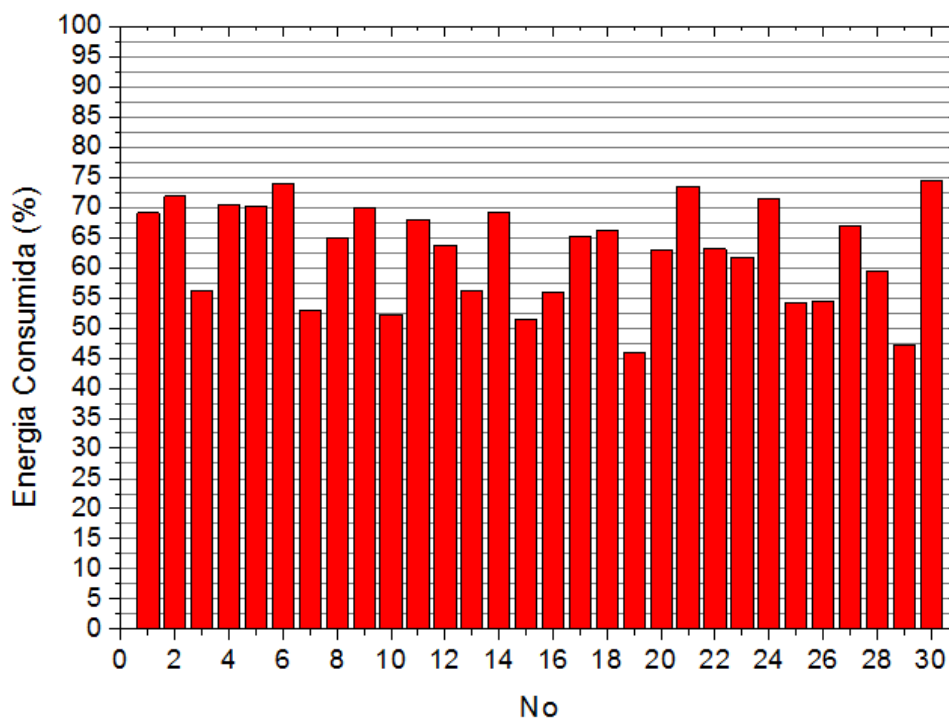
Figura 14 – Energia restante em cada nó da rede.



Fonte: Autor.

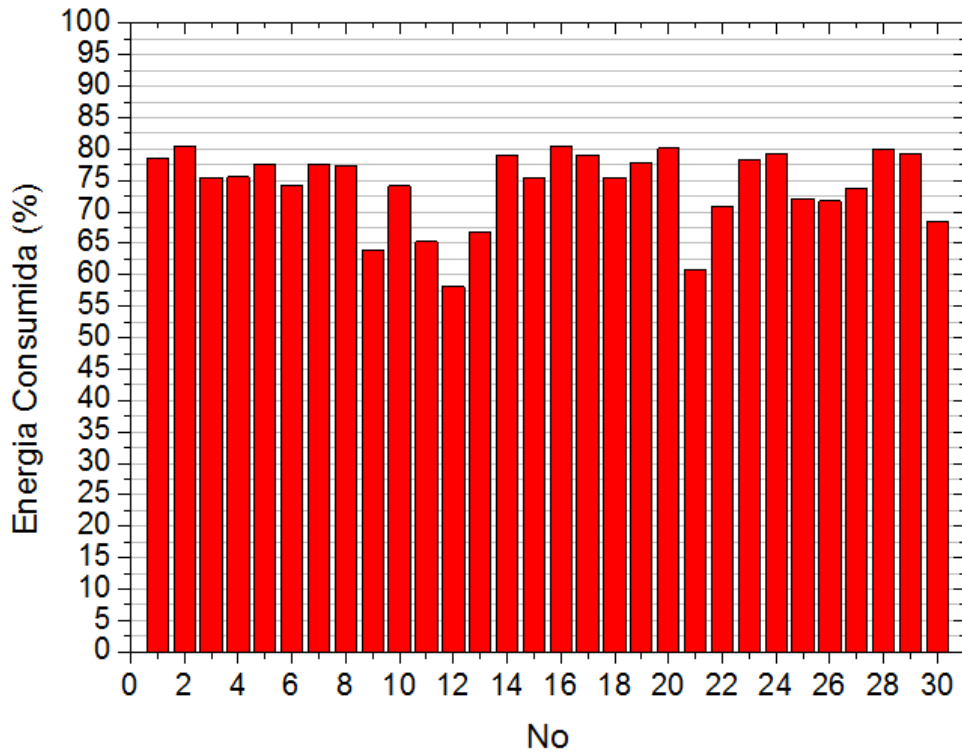
próximos do concentrador, e também tendem a ser nós intermediários mais vezes que os nós que se encontram mais distantes.

Figura 15 – Energia Consumida x Nó (Protocolo AODV).



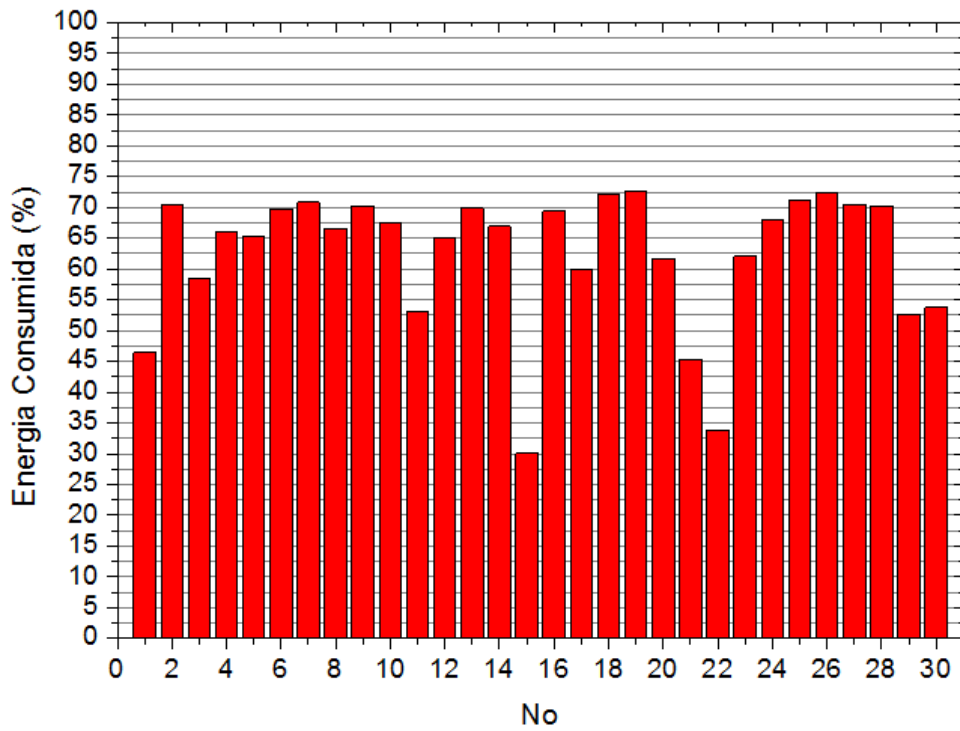
Fonte: Autor.

Figura 16 – Energia Consumida x Nó (Protocolo DSDV).



Fonte: Autor.

Figura 17 – Energia Consumida x Nó (Protocolo DSR).



Fonte: Autor.

### 4.5.2 Intervalo de Transmissão de 10 Minutos

A Tabela 10 demonstra os resultados obtidos no segundo cenário de simulação, onde os dados foram transmitidos em intervalos de 10 minutos. Analisando os resultados da Tabela 10 pode-se ver que o protocolo DSR apresentou apenas uma melhor PDR em relação aos protocolos AODV e DSDV.

O protocolo AODV apresentou maior *throughput*, menor *overhead* e consumo médio da bateria, isto é, para esse cenário o protocolo AODV iria fazer com que a bateria durasse mais tempo que os outros protocolos. Além disso, o protocolo apresentou menor atraso médio fim-a-fim.

O protocolo DSDV apresentou os piores resultados para esse cenário. Os resultados demonstram que um protocolo de roteamento proativo não apresenta um bom resultado para a aplicação em redes de sensores sem fio para monitoramento para pecuária.

Tabela 10 – Resultados do Segundo Cenário.

	<b>Protocolo de Roteamento</b>		
	AODV	DSR	DSDV
<b>Pacotes Enviados</b>	9918	9543	13820
<b>Pacotes Recebidos</b>	9878	9514	8843
<b>PDR (%)</b>	99,59	99,69	63,98
<b>Throughput (kbps)</b>	32,68	31,41	27,79
<b>Overhead</b>	0,519	0,529	1,543
<b>Atraso Médio Fim-a-Fim (ms)</b>	156,222	168,211	186,981
<b>Consumo Médio da Bateria (%)</b>	31,02	31,47	35,73

Fazendo a estimativa de duração da rede para bateria de 5292J com os resultados de consumo médio acima, a rede vai durar 35,52, 35,02 e 30,84 dias para os protocolos AODV, DSR e DSDV, respectivamente.

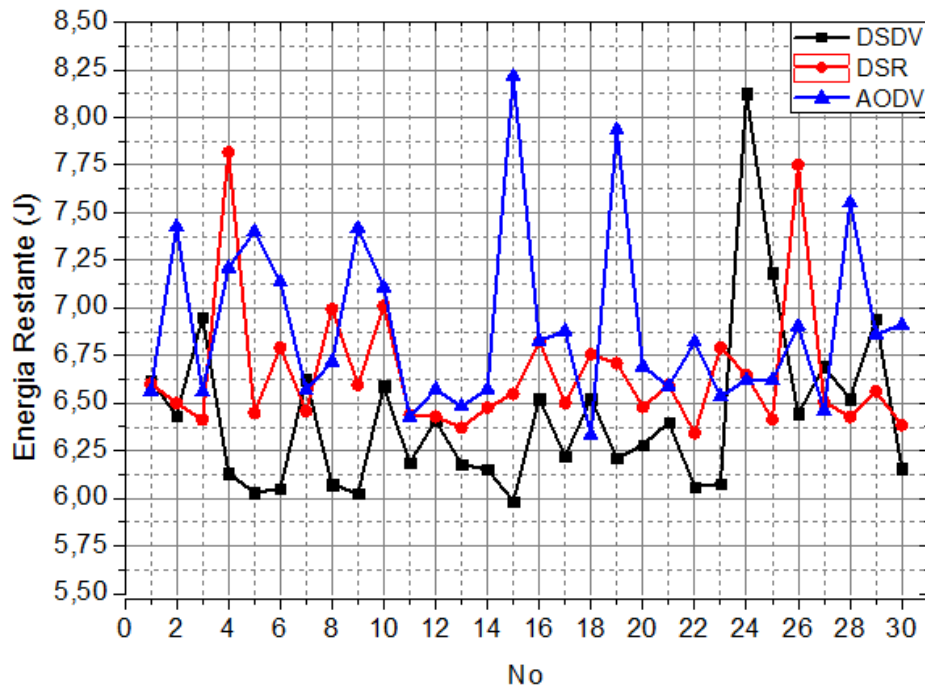
A Figura 18 demonstra o gráfico da energia restante em cada nó da rede para transmissão a cada 10 minutos entre os três protocolos. Como pode-se analisar o protocolo AODV apresentou a maior energia restante para seus nós.

As Figuras 19, 20 e 21 apresentam a porcentagem de energia consumida por cada nó da rede para seus respectivos protocolos de roteamento. Conforme explicado anteriormente, essas diferenças de energia consumida entre os nós de mesmo protocolo ocorre porque os nós se movimentam de maneira aleatória.

## 4.6 Influência da Quantidade de Nós na RSSF

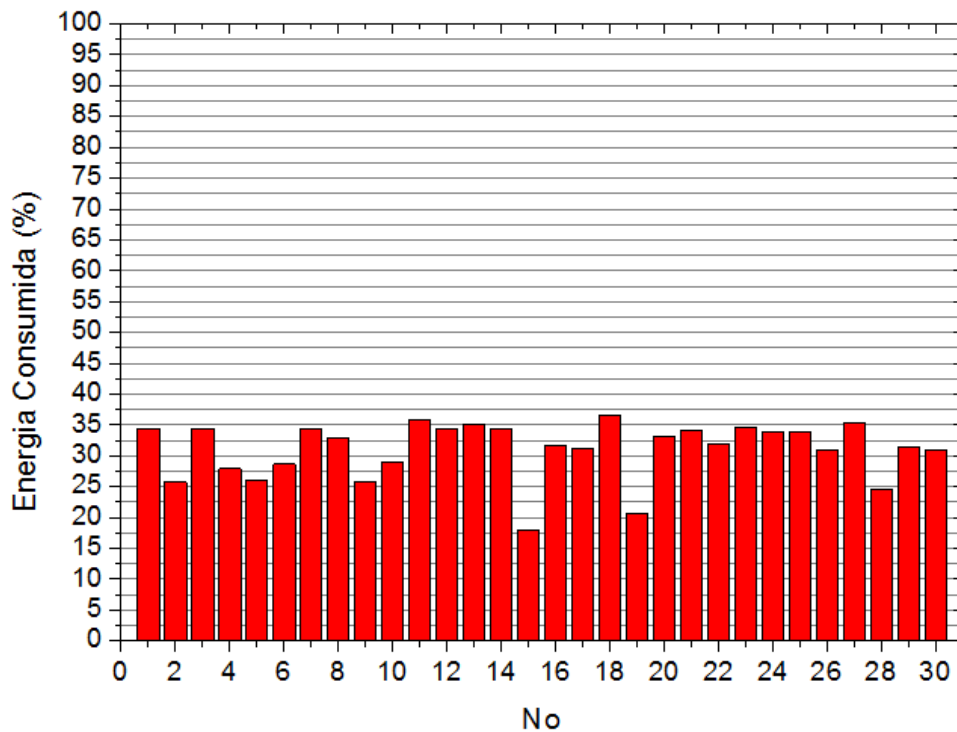
Por último, alterou-se a quantidade de nós sensores da rede sensor sem fio do primeiro cenário (Seção 4.5.1). Não foi alterada nenhuma outra característica do cenário

Figura 18 – Gráfico Vida Útil vs Tempo de Atualização do Módulo GPS.



Fonte: Autor.

Figura 19 – Energia Consumida x Nó (Protocolo AODV).

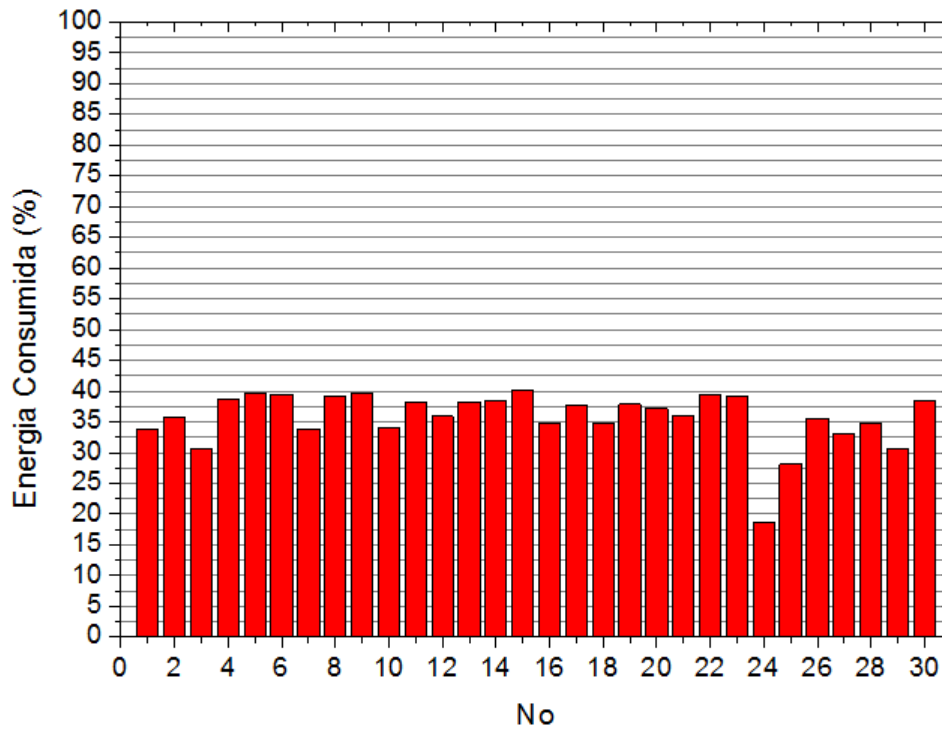


Fonte: Autor.

sem ser a quantidade de nós para verificar o impacto da variação dos nós em um mesmo cenário. Então, foram simulados dois cenários distintos, primeiramente com 15 nós e posteriormente com 45 nós, que representa tanto uma diminuição ou aumento de 50% dos

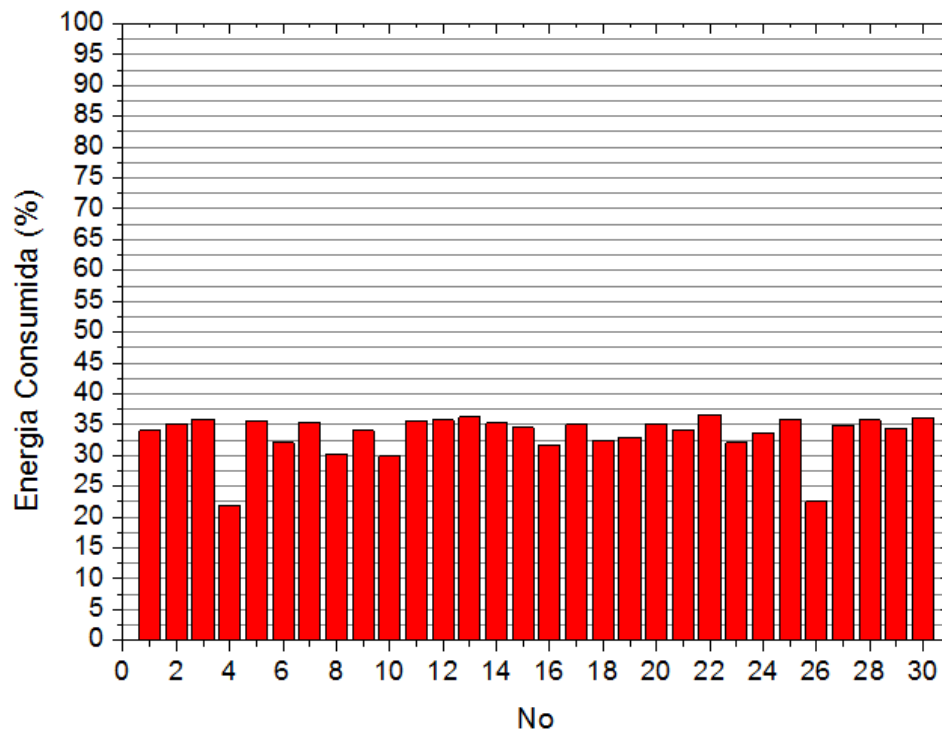


Figura 20 – Energia Consumida x Nó (Protocolo DSDV).



Fonte: Autor.

Figura 21 – Energia Consumida x Nó (Protocolo DSR).



Fonte: Autor.

nós no cenário. A Tabela 11 apresenta os valores obtidos nas simulações.

No cenário com 15 nós, o protocolo AODV apresentou os melhores resultados em

todos os parâmetros de avaliação. Entretanto, para o cenário com 45 nós, o protocolo DSR apresentou os melhores resultados, exceto com o *throughput* menor que os demais protocolos. Esses resultados ocorreram porque independentemente de ambos os protocolos serem reativos (sob demanda), eles apresentam diferenças no modo de como funciona o armazenamento das rotas. O protocolo DSR armazena múltiplas rotas para um só destino, então, com o número de nós elevados, o protocolo possui múltiplas rotas para escolher a mais curta para fazer o envio, e assim consegue melhores resultados. Por outro lado, o AODV utiliza em sua tabela no máximo uma entrada para cada destino, ou seja, não contém a rota inteira até um determinado nó. Isso faz com que o protocolo DSR seja melhor em redes com grandes números de nós, enquanto o AODV apresenta o melhor desempenho para redes com pequeno número de nós.

Comparando os resultados da Tabela 11 com os resultados da Tabela 9, pode-se observar uma grande variação nos resultados de consumo médio de energia da rede. Conforme aumenta o número de nós sensores da rede sensor sem fio, maior é o consumo médio da rede. Essa variação no resultado consumo médio da rede está relacionado a complexidade e a escalabilidade da rede, pois conforme vai aumentando o número de nós da rede, maior é a complexidade e a escalabilidade.

Tabela 11 – Resultados da Influência da Quantidade de Nós.

Quantidade de Nós	15 Nós			45 Nós		
	Protocolo de Roteamento	AODV	DSDV	DSR	AODV	DSDV
Pacotes Enviados	8993	10441	6667	25132	39588	25338
Pacotes Recebidos	8922	8297	6594	24923	25688	25174
PDR (%)	99,21	79,46	98,90	99,16	64,88	99,35
Throughput (kbps)	24,14	22,46	22,40	66,08	66,46	65,84
Overhead	0,338	0,58	0,573	0,998	1,04	0,903
Atraso Médio Fim-a-Fim (ms)	157,84	273,79	168,97	184,65	286,68	158,57
Consumo Médio da Bateria (%)	24,78	26,85	25,40	85,58	86,86	80,65

Fazendo a estimativa de duração da rede para bateria de 5292J com os resultados de consumo médio acima, a rede com 15 nós sensores vai durar 44,08, 41,04 e 43,38 dias para os protocolos AODV, DSDV e DSR, respectivamente. Para 45 nós sensores a rede vai durar 12,87, 12,68 e 13,66 dias para os protocolos AODV, DSDV e DSR, respectivamente.

Também foi realizada uma simulação considerando a bateria de 5292J para comprovar que é válida a estimativa de duração da rede sensor com a bateria de 10J para as simulações. Foi utilizado o cenário com 15 nós sensores com o protocolo de comunicação AODV para fazer a comparação dos resultados. Com o tempo de simulação de 30 minutos, a rede teve um consumo médio de energia de 0,05%, o que representa a duração da rede por 41,66 dias. Comparando este resultado com o resultado de 44,08 dias apresentado anteriormente para o mesmo cenário, o resultado mostra que a aproximação com a bateria de 10J é válida, visto que, acontecem variações de resultados de uma simulação para outra,

e também que a movimentação aleatória dos nós tem influência de uma simulação para outra.

## 4.7 Comentários

Neste capítulo foram apresentados as características e funcionamento do software de simulação NS-2, os cenários e ambientes de simulações, as métricas de desempenho consideradas para validar os protocolos de comunicação e os resultados obtidos após as simulações.

## 5 Conclusão

Esse trabalho proporcionou simular cenários de redes de sensores sem fio, com o objetivo de verificar a influência dos parâmetros de configuração dos nós em relação ao aumento do tempo de vida útil da rede, além de buscar o melhor protocolo de roteamento para uma RSSF para monitoramento para pecuária. Além disso, outro enfoque em que o trabalho foi direcionado, refere-se à importância em utilizar esse tipo de metodologia (simulação) para embasar um direcionamento estratégico na tomada de uma decisão em uma rede de sensores.

Comparando os resultados obtidos neste trabalho através dos cenários de simulação, pode-se concluir que o protocolo que irá apresentar melhor desempenho depende quase exclusivamente do cenário, onde o tempo de transmissão e a quantidade de nós na rede sensor irá influenciar no resultado. O protocolo DSR apresentou um melhor resultado para o cenário em que o tempo entre as transmissões é menor, e o protocolo AODV apresentou um melhor resultado para um maior tempo entre as transmissões. Os resultados deixam claro que os protocolos reativos (sob demanda) apresentam os melhores resultados para redes de sensores sem fio. Em nenhum cenário o protocolo DSDV apresentou melhor desempenho que os demais protocolos.

O software de simulação utilizado, o NS-2, é bastante difundido no meio acadêmico. No entanto, possui certa complexidade, requerendo um conhecimento prévio para a sua correta utilização. Isso ocorre pelo fato de ele apresentar algumas dificuldades em sua configuração, principalmente quando à compilação de seu código fonte. Alguns passos para compilação não são bem documentados, o que torna o processo de instalação e configuração difícil.

Em resumo, este trabalho demonstrou que os softwares de simulação são ferramentas importantes para a tomada de decisão no nível gerencial da organização, já que permitem definir, testar e avaliar de forma antecipada, quais serão os equipamentos que melhor se adaptarão à realidade do ambiente, bem como, a possibilidade de prever a quantidade e o modelo a ser utilizado, evitando investimentos desnecessários ou insuficientes.

### 5.1 Trabalhos Futuros

Como trabalho futuro pode-se citar o estudo e implementação do padrão IEEE 802.15.4 na camada MAC do NS-2. Esse padrão especifica a camada física e efetua o controle de acesso para redes de sensores sem fio com baixas taxas de transmissão e baixo consumo energético. Outro estudo, será a influência do aumento da quantidade

de concentradores na RSSF para estes cenários. Também está como trabalho futuro, a implementação prática de uma rede de sensores sem fio para um pequeno número de nós e um concentrador.

# Referências

- AGRICULTURA, M. da. *Bovinos e Bubalinos*. 2016. <<http://www.agricultura.gov.br/animal/especies/bovinos-e-bubalinos>>. [Online; accessed 10-November-2016]. Citado na página 14.
- AKYILDIZ, I. et al. A survey on sensor networks. In: *Communications Magazine, IEEE*, vol.40, no.8, pp. 102-114. [S.l.: s.n.], 2002. Citado 6 vezes nas páginas 16, 17, 18, 19, 20 e 21.
- ARDUINO. *Arduino Pro Mini*. 2016. <<http://www.arduino.cc/en/Main/ArduinoBoardProMini>>. [Online; accessed 10-November-2016]. Citado 3 vezes nas páginas 30, 31 e 32.
- BERKELEY, U. et al. *The ns Manual (formerly ns Notes and Documentation)*. [S.l.]: The VINT Project, 2011. Citado 2 vezes nas páginas 39 e 41.
- BRASIL, P. *Produção agropecuária brasileira de 2015 superou 463 bilhões de reais*. 2015. <<http://www.agricultura.gov.br/animal/especies/bovinos-e-bubalinos>>. [Online; accessed 10-November-2016]. Citado na página 13.
- BRASIL, P. *Rebanho bovinos brasileiro cresce e chega a 121,3 milhões de cabeça de gado*. 2015. <<http://www.brasil.gov.br/economia-e-emprego/2015/10/rebanho-bovino-brasileiro-cresce-e-chega-a-212-3-milhoes-de-cabecas-de-gado>>. [Online; accessed 10-November-2016]. Citado na página 13.
- BURATTI, C. e. a. *Sensor Networks with IEEE 802.15.4: Systems Distributed Processing, MAC, and Connectivity*. [S.l.: s.n.], 2011. Citado 3 vezes nas páginas 16, 17 e 18.
- ELECTRONICS, V. *Data Sheet VK2828U7G5LF G-Mouse*. V1.0. [S.l.], 2012. Citado 4 vezes nas páginas 9, 10, 33 e 34.
- ENERGIZER. Product datasheet energizer nh22-175. Citado na página 43.
- HEREK, T. A. *ANÁLISE COMPARATIVA DE DESEMPENHO DOS PROTOCOLOS DE ROTEAMENTO EM REDES MÓVEIS AD HOC*. Dissertação (Mestrado) — UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ, 2011. Citado 4 vezes nas páginas 22, 23, 27 e 28.
- HOBLOS, G.; STAROSWIECKI, M.; AITOUICHE. Optimal design of fault tolerant sensor networks. 2000. Citado na página 18.
- LIMA, R. de et al. Satellite telemetry system for pollution detection on insulator strings of high-voltage transmission lines. *SBMO/IEEE MTT-S International*, 2009. Citado na página 16.
- MUCHA, P. M. de A. *O Uso de Redes Sensores Sem Fio para Getão de TI em Ambientes Agrícolas*. Dissertação (Mestrado) — Universidade Federal de Santa Maria, 2013. Citado 2 vezes nas páginas 36 e 37.

NETO, J. A. T. C. *Análise de Protocolos de Roteamento Unicast em Redes Ad Hoc Móveis Baseada em um Modelo Realístico de Bateria*. Dissertação (Mestrado) — Universidade Federal de Campina Grande, 2011. Citado 3 vezes nas páginas 24, 25 e 26.

RIO., P. *Protocolos de Roteamento para MANET*. [S.l.: s.n.], 2010. Citado 4 vezes nas páginas 24, 25, 26 e 27.

SANTOS, L. R. dos. *Roteamento de Redes Sensores Sem Fio com Base em Aprendizagem por Reforço*. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, 2011. Citado na página 36.

SEMICONDUCTOR, N. *nRF905 Single chip 433/868/915MHz Transceiver*. [S.l.], 2008. Citado 3 vezes nas páginas 31, 32 e 33.

SHARMA, R.; GUPTA, K. Comparison based performance analysis of udp/cbr and tcp/ftp traffic under aodv routing protocol in manet. *International Journal of Computer Applications*, 2012. Citado 2 vezes nas páginas 41 e 42.

SILVA, L. H. de S. Desenvolvimento de uma rede de sensores sem fio utilizando zigbee para aplicações diversas. In: . [S.l.]: Trabalho de Conclusão de Curso Engenharia da Computação, 2011. Citado 6 vezes nas páginas 13, 15, 16, 17, 18 e 21.

VERONA, A. B. *SIMULAÇÃO E ANÁLISE DE REDES DE SENSORES SEM FIO APLICADAS À VITICULTURA*. Dissertação (Mestrado) — Universidade Estadual de Maringá, 2010. Citado 2 vezes nas páginas 36 e 37.

ZHAO, F.; GUIBAS, L. *Wireless Sensor Networks: An Information Processing Approach*. [S.l.: s.n.], 2004. Citado 3 vezes nas páginas 15, 19 e 20.