

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS

**CENTRO DE CIÊNCIAS EXATAS, AMBIENTAIS E DE
TECNOLOGIA**

VITOR QUEIROZ PEREIRA

**ESTIMATIVA DA PER, PROTOCOLO PARA
COLETA DE RSSI E DETERMINAÇÃO DE
MELHORES ROTAS EM RSSF**

**PUC-Campinas
2016**

VITOR QUEIROZ PEREIRA

**ESTIMATIVA DA PER, PROTOCOLO PARA
COLETA DE RSSI E DETERMINAÇÃO DE
MELHORES ROTAS EM RSSF**

Dissertação apresentada como exigência para obtenção do Título de Mestre em Engenharia Elétrica, ao Programa de Pós-graduação em Gestão de Redes de Telecomunicações, do Centro de Ciências Exatas, Ambientais e de Tecnologia da Pontifícia Universidade Católica de Campinas.

Orientador: Prof. Dr. Omar Carvalho Branquinho

**PUC-Campinas
2016**

Ficha Catalográfica
Elaborada pelo Sistema de Bibliotecas e
Informação - SBI - PUC-Campinas

t621.3851
P436e Pereira, Vitor Queiroz.
 Estimativa da PER, protocolo para coleta de RSSI e determinação
 de melhores rotas em RSSF / Vitor Queiroz Pereira. - Campinas: PUC-
 Campinas, 2016.
 100p.

 Orientador: Omar Carvalho Branquinho.
 Dissertação (mestrado) – Pontifícia Universidade Católica de Cam-
 pinas, Centro de Ciências Exatas, Ambientais e de Tecnologias, Pós-
 Graduação em Engenharia Elétrica.
 Inclui bibliografia.

 1. Redes de sensores sem fio. 2. Gerência. 3. Engenharia elétrica.
 I. Branquinho, Omar Carvalho. II. Pontifícia Universidade Católica de
 Campinas. Centro de Ciências Exatas, Ambientais e de Tecnologias.
 Pós-Graduação em Engenharia Elétrica. III. Título.

22.ed. CDD – t621.3851

VITOR QUEIROZ PEREIRA

**ESTIMATIVA DA PER, PROTOCOLO DE COLETA DE
RSSI E DETERMINAÇÃO DE MELHORES ROTAS EM
RSSF**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro de Ciências Exatas, Ambientais e de Tecnologias da Pontifícia Universidade Católica de Campinas como requisito parcial para obtenção do título de Mestre em Gestão de Redes de Telecomunicações.

Área de Concentração: Gerência de Redes de Teleinformática. Orientador: Prof. Dr. Omar Carvalho Branquinho

Dissertação defendida e aprovada em 23 de junho de 2016 pela Comissão Examinadora constituída dos seguintes professores:



Prof. Dr. Omar Carvalho Branquinho
Orientador da Dissertação e Presidente da Comissão Examinadora
Pontifícia Universidade Católica de Campinas



Profa. Dra. Indayara Bertoldi Martins
Pontifícia Universidade Católica de Campinas



Prof. Dr. Paulo Cafieri
UNICAMP

**Aos meus pais, pelo apoio
incondicional.**

“Diante da vastidão do tempo e da imensidão do universo, é um imenso prazer para mim dividir um planeta e uma época com você.”

Carl Sagan

Resumo

PEREIRA, Vitor Queiroz. **ESTIMATIVA DA PER, PROTOCOLO DE COLETA DE RSSI E DETERMINAÇÃO DE MELHORES ROTAS**. 2016. Dissertação (mestrado em Engenharia Elétrica) – Programa de Pós-Graduação em Gestão de Redes de Telecomunicações, Pontifícia Universidade Católica de Campinas, Campinas, 2016.

As características das Redes de Sensores Sem Fio (RSSF) fazem desta tecnologia uma solução para diversas finalidades, inclusive quando se trata de Internet das Coisas (IoT). Formada por nós sensores, este tipo de rede é responsável por monitorar e atuar em diferentes aspectos do ambiente, como medir a temperatura ou acionar um exaustor. Entretanto, a comunicação sem fio é vulnerável as mudanças do ambiente em que está inserida, podendo comprometer a eficácia da comunicação. Neste sentido, este trabalho apresenta uma proposta de metodologia para determinação de melhores rotas baseada em uma estimativa da PER pela RSSI e um protocolo para a coleta das RSSIs da rede. A estimativa da PER foi desenvolvida por meio de testes experimentais com a modulação FSK. O protocolo de coleta foi implementado em uma rede real e as RSSIs utilizadas na determinação de melhores rotas. A metodologia para atribuição de rotas foi desenvolvida baseada na classificação dos nós em função de sua criticidade.

Palavras-chave: RSSF, Redes de Sensores Sem Fio, IoT, Internet das Coisas, RSSI, Melhores Rotas, PER, FSK.

Abstract

PEREIRA, Vitor Queiroz. **ESTIMATE PER, PROTOCOL FOR RSSI COLLECTION AND DETERMINATION OF BEST ROUTES**. 2016. Dissertation (Master in Electrical Engineering) - Graduate Program in Telecommunications Network Management, Pontifícia Universidade Católica de Campinas, Campinas, 2016.

The characteristics of the Sensors Wireless Networks (WSNs) make this technology a solution for a variety of purposes, including when it comes to Internet of Things (IoT). Formed by sensor nodes, this type of network is responsible for monitoring and acting on various aspects of the environment to measure temperature or trigger an exhaust fan. However, wireless communication is vulnerable environment changes in which it operates, which could compromise the effectiveness of communication. Thus, this work presents a proposal for a methodology to determine optimal routes based on an estimate of the PER by the RSSI and a protocol for the collection of network RSSIs. The estimated PER was developed by experimental tests with FSK modulation. The collection protocol was implemented in a real network and RSSIs used to determine the best routes. The methodology for allocation of routes was developed with the classification premise of us according to the criticality of your application into two groups, high and low criticality.

Keywords: WSN, Sensor Networks Wireless , IoT , Internet of Things, RSSI , Best Routes, PER , FSK .

Lista de Figuras

FIGURA 1 – CONEXÃO DA RSSF NA INTERNET	14
FIGURA 2 – PILHA ZIGBEE COM IEEE902.15.4	18
FIGURA 3 – ARQUITETURA DE REDE DA WIRELESSHART	19
FIGURA 4 - TOPOLOGIAS	23
FIGURA 5 – GRÁFICO Eb/No X BER	24
FIGURA 6 – CRESCIMENTO DA REDE	25
FIGURA 7 – ENVIOS DE PACOTES POR TOPOLOGIA.....	29
FIGURA 8 – GRÁFICO PER X LATÊNCIA	33
FIGURA 9 – GRÁFICO TOTAL DE NÓS X LATÊNCIA.....	33
FIGURA 10 – GRÁFICO Eb/No X PER E BER	34
FIGURA 11 – GRÁFICO RSSI X ESTIMATIVA TEÓRICA DA PER.....	37
FIGURA 12 – GERENTE NA RSSF	39
FIGURA 13 – FLUXOGRAMA DO PROTOCOLO DE COLETA PARA OS NÓS SENSORES.....	44
FIGURA 14 – PSEUDOCÓDIGO DO PROTOCOLO DE COLETA PARA OS NÓS SENSORES.....	45
FIGURA 15 – DIAGRAMA TEMPORAL DO PROTOCOLO DE COLETA	46
FIGURA 16 – FLUXOGRAMA DO PROTOCOLO DE COLETA PARA A BASE	47
FIGURA 17 – PSEUDOCÓDIGO DO PROTOCOLO DE COLETA 2.....	49
FIGURA 18 – FLUXOGRAMA DO PROTOCOLO DE COLETA PARA OS NÓS SENSORES 2	50
FIGURA 19 – FLUXOGRAMA DO PROTOCOLO DE COLETA PARA A BASE	51
FIGURA 20 – BE900.....	59
FIGURA 21 – DK102	60
FIGURA 22 - UARTSBEE	60
FIGURA 23 – ATENUADORES FIXOS	61
FIGURA 24 – BANCADA DE TESTES PARA AFERIÇÃO DA PER.....	67
FIGURA 25 – RSSF DA IMPLEMENTAÇÃO DO PROTOCOLO DE COLETA.....	68
FIGURA 26 – MAPA DA DISTRIBUIÇÃO DE NÓS.....	69
FIGURA 27 – CENÁRIO IMPLEMENTADO	70
FIGURA 28 – RSSI X NORMAL E RESULTADOS EXPERIMENTAIS.....	72
FIGURA 29 – PACOTES ENVIADOS X PACOTES PERDIDOS.....	72
FIGURA 30 – PACOTES ENVIADOS X PER	73
FIGURA 31 – PER X Eb/No E Eb/No ESTIMADO	75
FIGURA 32 – RSSI X PER ESTIMADA E PER MEDIDA	75
FIGURA 33 – MELHORES ROTAS.....	79
FIGURA 34 – MELHOR ROTA, BAIXA CRITICIDADE.....	81

Lista de Tabelas

TABELA 1 – CATEGORIAS DE CRITICIDADE POR APLICAÇÃO	21
TABELA 2 – MATRIZ DA EXISTÊNCIA DE RETAS	26
TABELA 3 – ROTAS DO PROTOCOLO DE COLETA, 8 NÓS	41
TABELA 4 - ROTAS DO PROTOCOLO DE COLETA, 7 NÓS.....	41
TABELA 5 - ROTAS DO PROTOCOLO DE COLETA, 9 NÓS.....	43
TABELA 6 – ROTAS DO PROTOCOLO DE COLETA, 5 NÓS	50
TABELA 7 – ROTAS ALTERNATIVAS DO PROTOCOLO DE COLETA, 5 NÓS.....	52
TABELA 8 – RSSIS ENTRE NÓS, EXEMPLO.....	54
TABELA 9 – CLASSIFICAÇÃO DE RSSIS, BASE	54
TABELA 10 – CLASSIFICAÇÃO DE RSSIS, NÓ 4	55
TABELA 11 – CLASSIFICAÇÃO DE RSSIS, NÓ 2	56
TABELA 12 – CLASSIFICAÇÃO DE RSSIS, NÓ 3	57
TABELA 13 – MAPA DO PACOTE, FIRMWARE PARA AFERIÇÃO DA PER.....	63
TABELA 14 – MAPA DO PACOTE, FIRMWARE DO PROTOCOLO DE COLETA	64
TABELA 15 – RESULTADO DO EXPERIMENTO PARA A ESTIMATIVA DA PER	73
TABELA 16 – COMPARAÇÃO ENTRE RSSI E Eb/NO.....	74
TABELA 17 – DIFERENÇA ENTRE Eb/NO E RSSI AJUSTADA	74
TABELA 18 – RSSIS COLETADAS PELO PROTOCOLO DE COLETA.....	76
TABELA 19 – PERs ESTIMADAS PELAS RSSIS COLETADAS.....	77
TABELA 20 – CLASSIFICAÇÃO DE RSSIS, BASE	77
TABELA 21 – CLASSIFICAÇÃO DE RSSIS, NÓ 4	78
TABELA 22 – CLASSIFICAÇÃO DE RSSIS, NÓ 7	78
TABELA 23 – ROTAS ELEGÍVEIS, ALTA CRITICIDADE	79
TABELA 24 – LATÊNCIA RESULTANTE, PONTO A PONTO	80
TABELA 25 – LATÊNCIA RESULTANTE. 4 ANÉIS	80
TABELA 26 – LATÊNCIA RESULTANTE, 2 ANÉIS	81
TABELA 27 – LATÊNCIA RESULTANTE, ANEL	81

Sumário

1 Introdução.....	13
1.1 Objetivos e Contribuições	14
1.2 Organização do Trabalho	15
2 Revisão da Literatura	17
2.1 Tecnologias de RSSF.....	17
2.2 Criticidade de Eventos em Ambientes Industriais	20
2.3 Gerência de RSSF	22
2.4 Topologias.....	22
2.5 Modulação FSK.....	24
3 Análise de Tópicos em RSSF	25
3.1 Análise do Crescimento da Rede	25
3.2 Determinação de Melhores Rotas	28
3.2.1 Saltos X Topologia	28
3.2.2 PER X Saltos	30
3.2.3 Latência X PER	32
3.3 Estimativa da PER pela RSSI	34
3.3.1 Estimativa Teórica	35
4 Proposta	38
4.1 Estimativa da PER em Função da RSSI	38
4.2 Protocolo Para Coleta de Dados	38
4.2.1 Destino do Pacote Determinado pelo Nó Sensor de Envio.....	43
4.2.2 Destino do Pacote Calculado na Recepção	47
4.2.3 Atribuição de Rotas Alternativas.....	52
4.3 Determinação de Melhores Rotas	53
4.3.1 Aplicações de Maior Criticidade.....	53
4.3.2 Aplicações de Menor Criticidade	55
5 Material e Métodos.....	58
5.1 Hardware	58
5.2 Firmware	61
5.2.1 Firmware para Aferição da PER.....	62
5.2.2 Firmware do Protocolo de Coleta de RSSIs.....	63
5.3 Softwares	65
5.3.1 Software para Aferição da PER	65

5.3.2 Software Gerente do Protocolo de Coleta de RSSIs.....	66
5.4 Bancada de Testes para Aferição da PER pela RSSI	66
5.5 Implementação do Protocolo de Coleta	67
5.5.1 Cenário de Teste	68
6 Análise de Resultados	71
6.1 Aferição da PER.....	71
6.1.1 Validação dos Resultados.....	71
6.1.2 Estimativa da PER em Função da RSSI.....	73
6.2 Coleta de RSSIs.....	76
6.3 Rotas criadas para aplicações de maior criticidade	77
6.4 Rotas criadas para aplicações de menor criticidade.....	80
7 Conclusão	82
Referências.....	84
Anexo A – Firmware Radiuino – Aferição da PER – Transmissor	86
Anexo B – Firmware Radiuino – Aferição da PER – Receptor	88
Anexo C – Firmware Radiuino – Protocolo para Coleta de RSSI	89
Anexo D – Software Phyton – Aferição da PER.....	93
Anexo E – Software Phyton – Coleta de RSSIs	97

1 Introdução

Rede de Sensores Sem Fio (RSSF) é um tema que vem sendo tratado extensamente na literatura (AKYILDIZ, 2011; KARL, 2005), principalmente em estudos que envolvem simulações. Em geral, na literatura se propõem RSSF com nós sensores dotados de capacidade de processamento suficiente para implementação de estratégias para atribuição de rotas. Estas estratégias implicam na autonomia dos nós sensores na busca de melhores caminhos. Embora interessante do ponto de vista acadêmico, esta estratégia não tem obtido resultados práticos tão interessantes, uma vez que os algoritmos dependem do tipo de ambiente e de como a rede está montada. Sendo assim é possível classificar as estratégias de busca de melhores rotas como sendo uma forma distribuída, sem um elemento central que coordene a rede criando rotas e gerenciando o funcionamento da rede.

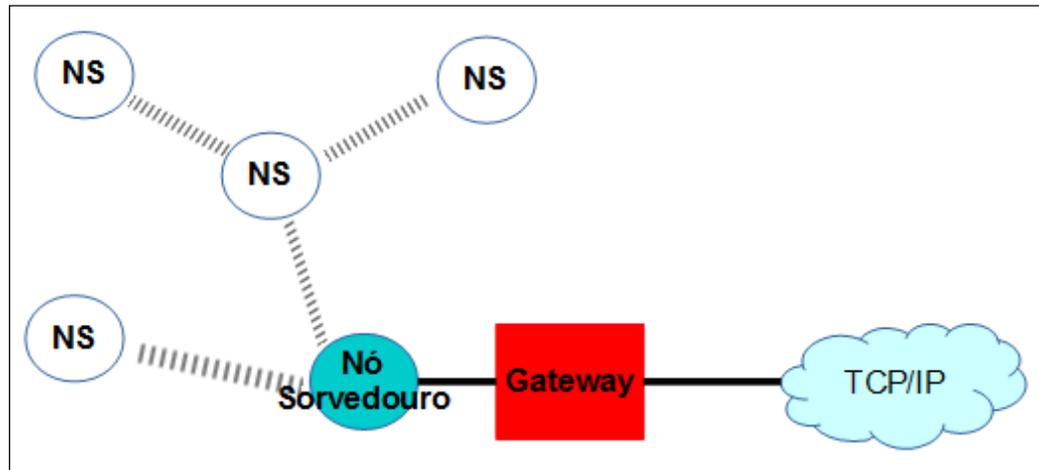
Este trabalho apresenta uma alternativa com a existência de um gerente da rede dedicado a atribuição de rotas e gerenciamento do funcionamento da rede. Esta é uma estratégia centralizada, na qual os nós sensores são acessados por solicitação do gerente. Assim, os nós sensores sem fio não realizam busca de melhores rotas e atendem ao que estabelece o gerente, além de não ficarem responsáveis pela gerência da rede.

Este é um dos pontos chave para o insucesso de muitas RSSF que consideram que a independência dos nós na busca de melhores caminhos é uma estratégia possível em qualquer cenário. Como consequência desta estratégia o custo tecnológico é muito alto, uma vez que o usuário final fica refém da tecnologia implementada, dependendo dos algoritmos implementados pelos fabricantes. Assim, embora exista padrões, como IEEE802.15.4[IEEE] e ZigBee[ZigBee], estes somente definem interface e não estratégia de implementação, que fica a cargo dos fabricantes criando diferentes soluções tanto de *hardware* quanto de *firmware*.

O contexto em que o trabalho está inserido é o da Internet das Coisas (IoT). Trata-se de uma área na qual busca-se conectar coisas à internet para monitoramento e/ou controle do que se deseja. A mobilidade fornecida por uma RSSF aliada aos custos reduzidos de implementação fazem desta uma solução

que tende a ser amplamente utilizada neste contexto (MADEIRA, 2013). Na Figura 1 é apresentado os elementos básicos de uma RSSF conectada à internet que permitem sua utilização em IoT (OLIVEIRA, 2016).

Figura 1 – Conexão da RSSF na Internet



Fonte: OLIVEIRA, 2016

Na figura, NS são os nós sensores. Nó Sorvedouro, também chamado de base ou *sink node*. *Gateway* é o responsável pela comunicação entre a RSSF e a internet, TCP/IP.

1.1 Objetivos e Contribuições

O trabalho propõe uma estratégia centralizada em que um gerente tem por função encontrar as rotas mais adequadas entre os nós sensores. Para criação destas rotas é necessário definir parâmetros que indiquem a qualidade da rota, como a intensidade de sinal recebida e a taxa de erro. Com estas informações é possível determinar os melhores caminhos.

Para realizar a coleta da intensidade de sinal recebida é necessário uma estratégia de busca desta informação entre todos os sensores da rede para seja possível determinar as melhores rotas.

Com a estratégia centralizada o gerente da rede tem controle de como foi formada a rede e pode monitorar o comportamento da mesma e tomar decisão de redirecionar pacotes, quando necessário.

Outra característica é a possibilidade de criar estratégias de priorização do tráfego em função da criticidade da informação transportada.

Neste trabalho é proposto um algoritmo para busca da intensidade de sinal entre todos os nós da rede, uma das contribuições do trabalho. Para isto foram criados um *software* gerente e um *firmware* dos nós sensores para cumprir esta tarefa de busca da intensidade de sinal para posterior descoberta de melhores rotas.

1.2 Organização do Trabalho

No Capítulo 2 é feita uma revisão da literatura a respeito. São identificados trabalhos que apresentam estratégia de criação de RSSF, com diferentes abordagens quanto à formação da rede e gerência.

No Capítulo 3 é feita uma análise dos tópicos em RSSF. Neste capítulo é feita uma análise de como se dá o crescimento da. São analisadas as questões relacionadas com a determinação de melhores rotas como Saltos x Topologia, PER x Saltos e Latência x PER. Outra questão abordada neste capítulo é a estimativa da PER pela RSSI.

O Capítulo 4 apresenta a proposta do trabalho. Este capítulo se divide em estimativa da PER pela RSSI, protocolo para medida da RSSI entre todos os nós sensores e a determinação de melhores rotas.

O Capítulo 5 apresenta o material e os métodos utilizados para a implementação das estratégias propostas. É apresentado o firmware implementado nos nós sensores, o software de gerência da rede e o hardware utilizado nos testes realizados.

O Capítulo 6 apresenta os resultados obtidos em uma implementação real em um laboratório, com a criação de um cenário para avaliar o desempenho da estratégia.

O Capítulo 7 apresenta a conclusão do trabalho.

2 Revisão da Literatura

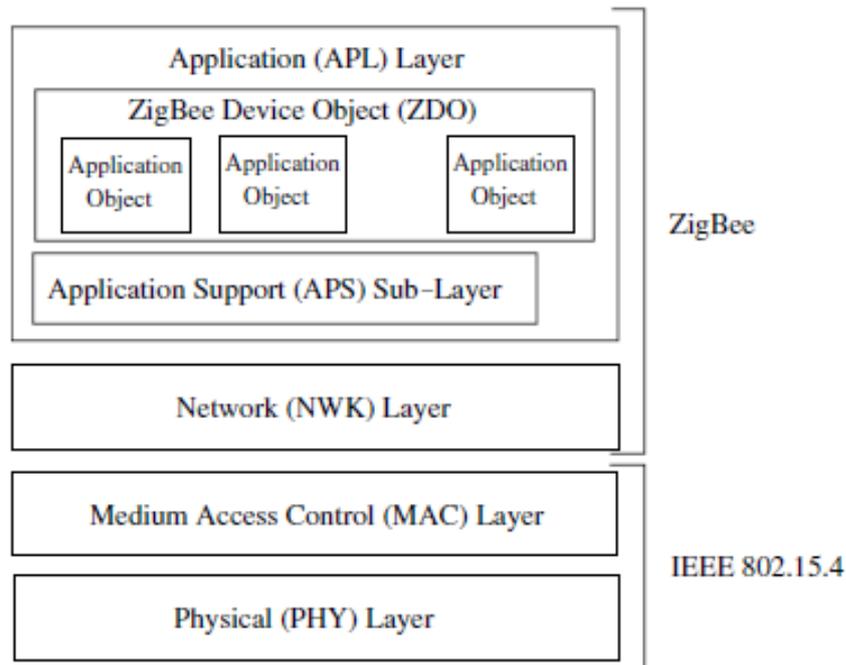
Uma revisão literária sobre algumas tecnologias, criticidade de aplicações em ambientes industriais, gerência em RSSF e a modulação FSK são tratadas neste capítulo.

2.1 Tecnologias de RSSF

A heterogeneidade das plataformas de nós sensores resulta em problemas na comunicação entre elas. Por isso, há um esforço pela padronização dos protocolos de comunicação em RSSF, segundo AKYLDIZ, 2010. O padrão IEEE 802.15.4 [IEEE] estabelece as características da comunicação dos transceptores em redes sem fio de baixas complexidade, potência de transmissão, consumo e taxa de comunicação. Este padrão define as características das camadas Física e MAC, fornecendo flexibilidade as camadas superiores. Define especificamente o espectro eletromagnético a ser utilizado, técnicas de comunicação sem fio e algoritmos da MAC. Isso permite que transceptores compatíveis se comuniquem mesmo sendo produzidos por diferentes fabricantes.

Outros padrões foram criados com o propósito de atender as necessidades de padronização nas camadas superiores. O padrão ZigBee[ZigBee], desenvolvido pela ZigBee Alliance, para atender a demanda em cinco principais áreas: automação residencial, *smart energy*, automação predial, serviços de telecomunicações e *personal health care*. Este padrão opera em conjunto com o IEEE 802.15.4, mas atuam em camadas distintas, como apresentado na Figura 2.

Figura 2 – Pilha ZigBee com IEEE902.15.4



Fonte: Akyldiz, 2011

As camadas Física e MAC são definidas pelo IEEE 802.15.4 enquanto que as camadas de Rede e Aplicação são definidas pelo ZigBee. A camada de Rede fornece funcionalidades de gestão para o funcionamento da rede, procedimentos para estabelecer uma nova rede e identificação e inserção de novos nós. Mecanismos de atribuição de endereçamentos e roteamento também são definidos nesta camada.

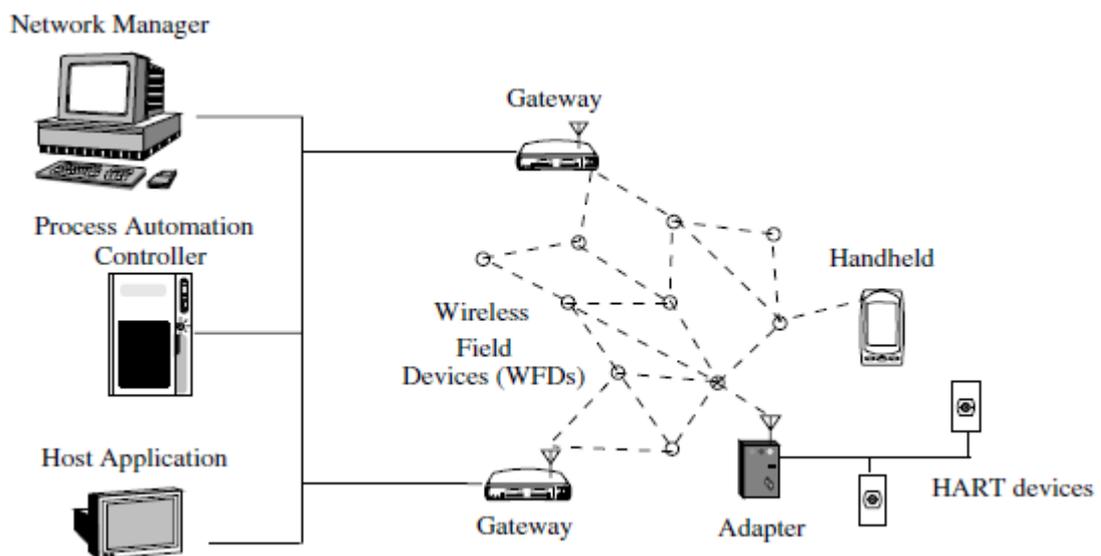
O padrão ZigBee divide a camada de aplicação em três: uma Subcamada de Aplicação, A ZigBee *Device Object* (ZDO) e o Objeto de Aplicação. As aplicações são implementadas no Objeto de Aplicação e se baseia nos requerimentos definidos pelo padrão. O ZDO define as funções fornecidas pelo dispositivo para o funcionamento da rede. Todos os dispositivos da rede, como o coordenador e o roteador são definidos pelo ZDO, incluindo as requisições de associação com a rede. A Subcamada de Aplicação fornece a capacidade de descobrimento de dispositivos vizinhos e armazenamento de suas funções.

Um dos desafios em RSSF é sua integração com a internet, uma vez que os protocolos baseados no IEEE 802.15.4 não são compatíveis com IP, exigindo um

gateway que realize a comunicação entre as redes. Para facilitar esta integração, a *Internet Engineering Task Force* (IETF) desenvolveu o padrão 6LoWPAN (*IPv6 over Low-power Wireless Personal Area Network*) [6LoWPAN]. Este padrão implementa a camada IPv6 sobre o IEEE 802.15.4, permitindo que qualquer dispositivo seja acessado pela internet. O maior desafio de implementação do IPv6 em uma RSSF é a estrutura de endereçamento de 40 bytes exigida por ele. Para isso são utilizadas técnicas de compressão que reduzem o tamanho do endereço, para cerca de 4 bytes, e uma camada de adaptação que permite a operação conjunta do IPv6 com as camadas do rádio. Este padrão tem como objetivo alcançar a interoperabilidade direta entre a internet e a RSSF.

Outro padrão bem difundido é o WirelessHART [WirelessHART], desenvolvido para atender as necessidades da indústria. Operando no modo de comunicação mestre/escravo e permitindo mais de dois mestres por escravo, a rede se caracteriza pela resiliência e confiabilidade na comunicação. A arquitetura de rede deste padrão está apresentada na Figura 3.

Figura 3 – Arquitetura de Rede da WirelessHART



Fonte: Akyldiz, 2011

Cinco tipos de elementos são definidos por este padrão: *Wireless Field Devices* (WFDs), são os sensores ou controladores conectados ao processo ou equipamento. Gateway fornece a interface entre a RSSF e a infraestrutura de rede cabeada. *Network Manager* mantém a operação da rede reservando os slots de

comunicação dos dispositivos, determina a tabela de roteamento e monitora o comportamento da rede. Os *WirelessHART Adapters* fornece a compatibilidade integrando os HART *devices* com a rede sem fio. *Handhelds* são equipados com transceptores que fornecem o acesso e interação com a rede sem fio.

A *Systems and Automation Society* (ISA) também desenvolveu um padrão para comunicação voltado para a indústria, o ISA 100 [ISA100]. Este apresenta o conceito de criticidade de aplicações em ambientes industriais. Diferenciando, assim, os nós por suas aplicações e relevâncias no processo industrial. Este tema será abordado no tópico seguinte.

2.2 Criticidade de Eventos em Ambientes Industriais

Dentre os padrões desenvolvidos, alguns são destinados a indústria, como WirelessHart e ISA100. Os benefícios das RSSF no ambiente industrial são discutidos em detalhe em [Industry 4.0], demonstrando que a evolução dos controles nos processos industriais se dará por este tipo de rede. São considerados três pontos principais:

- Aumento da confiabilidade: os sistemas cabeados podem estar expostos a condições extremas. Com RSSF é possível criar redundância;
- Melhora no controle: com uma RSSF podem ser inseridos mais dispositivos de monitoração para melhorar o controle;
- Diminuição de custo: estima-se que até 90% do custo das instalações das tecnologias de medida está em cabos, canaletas e construções. A RSSF torna economicamente viável a inclusão de novas aplicações.

A questão da utilização de RSSF no ambiente industrial tem como ponto crítico a qualidade da conexão rádio estabelecida entre os nós sensores. Tornando imprescindível a monitoração e gerência das RSSF (PEREIRA, 2015).

A qualidade de um enlace rádio, em especial em um ambiente industrial, exige a monitoração em tempo real com alguma estratégia que indique a necessidade de alteração dos recursos da rede. Mais especificamente os padrões

industriais relacionam a qualidade do enlace rádio com a criticidade dos processos envolvidos. Portanto, a qualidade do enlace rádio determina o impacto no processo que está sendo monitorado/controlado. Como exemplo, é apresentada uma tabela que relaciona as classes das aplicações envolvidas nos processos industriais. A Tabela 1, [ISA100b], apresenta esta classificação que utiliza uma escala de 0 a 5, em que o 0 é mais crítico e o 5 é o menos crítico.

Tabela 1 – Categorias de Criticidade por Aplicação

Categoria	Classe	Aplicação	Característica
Segurança	0	Ação de emergência	Sempre crítica
Controle	1	Controle regulador de malha fechada	Frequentemente crítica
	2	Controle de supervisão de malha fechada	Usualmente não crítica
	3	Controle de malha aberta	Participação Humana na malha
Monitoramento	4	Alerta	Consequências de curto prazo
	5	Armazenamento de dados – <i>downloading/uploading</i>	Sem consequências imediatas

Fonte: [ISA100b]

Esta tabela permite identificar que a utilização das RSSF em ambiente industrial necessita de uma análise da viabilidade, com a identificação de estratégias que permitam atender as especificações.

As RSSF em ambientes industriais terão sua configuração alterada para atender o quão crítico é o processo em que o enlace rádio está sendo utilizado. Portanto, a gerência de uma RSSF necessita de uma estrutura que utilize as informações de qualidade do enlace rádio para tomada de decisão.

2.3 Gerência de RSSF

A ISO (*International Standard Organization*) estabelece que a gerência é dividida em 5 áreas: configuração, desempenho, falha, segurança e bilhetagem (STALLINGS, 1999). Caracterização das áreas:

- Configuração - inicializar, desligar, manter, adicionar, atualizar e define parâmetros aos componentes.
- Desempenho - monitorar e avaliar o comportamento dos objetos gerenciados.
- Falha - detecção, isolamento e correção de operações anormais.
- Segurança - proteção relacionada ao controle de acesso aos recursos da rede e proteção da informação.
- Bilhetagem - contabilização de recursos e custos vinculados aos mesmos.

Este trabalho apresenta uma estratégia para avaliar a qualidade dos enlaces rádio pela gerência de desempenho, através da análise em tempo real da RSSI e a tomada de decisão para a alteração de parâmetros da RSSF pela gerência de configuração.

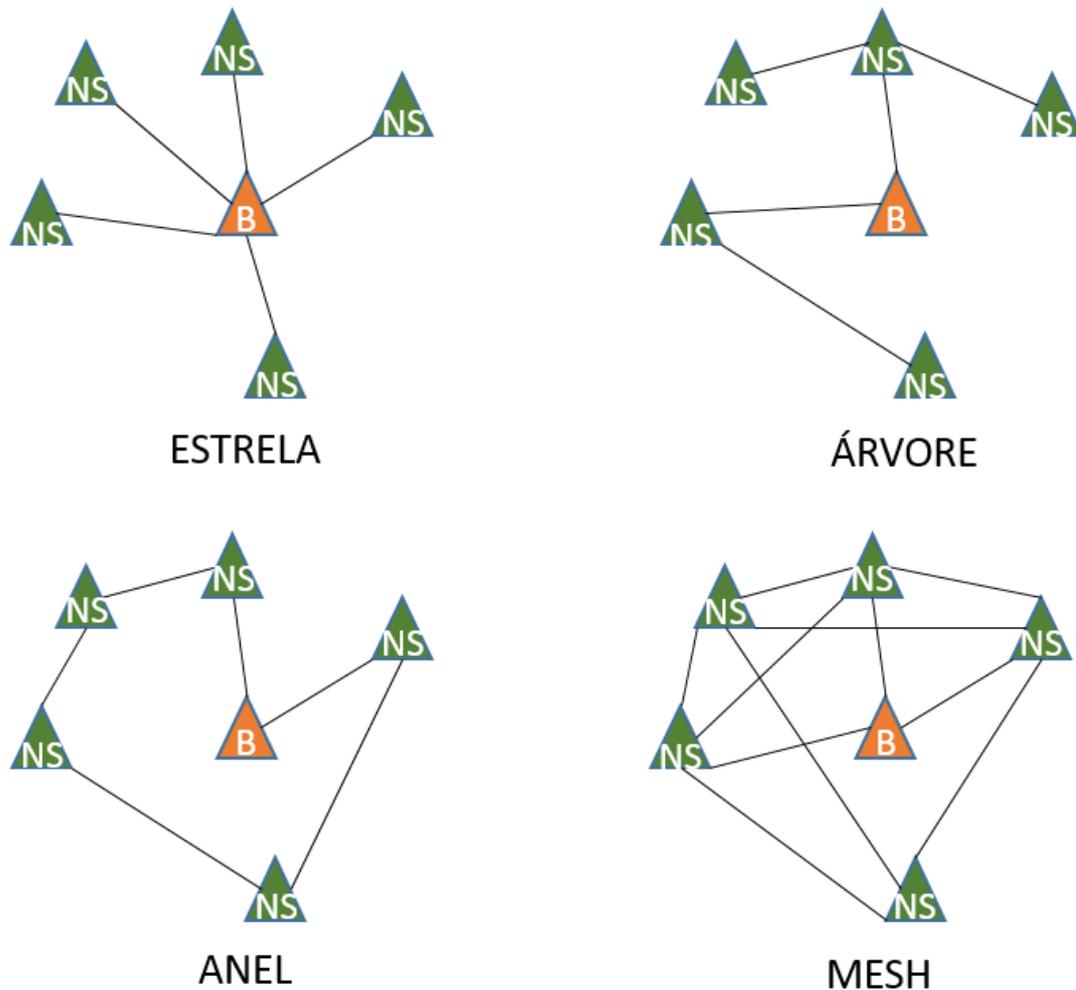
2.4 Topologias

As características físicas das Redes de Sensores Sem Fio permitem que se crie a topologia lógica da rede de várias maneiras. Cada topologia tem suas características que trazem vantagens e desvantagens (KARL, 2005). A mais simples de se implementar é topologia em estrela, neste caso somente a base troca pacotes ponto a ponto com os nós sensores, não existe comunicação entre eles. Outra arquitetura comum é a topologia em árvore, neste caso cria-se uma ramificação entre nós sensores formando uma estrutura hierárquica, isso permite uma cobertura de área maior que em estrela. A topologia em anel destaca-se pelo baixo número de links de comunicação da rede, diminuindo a quantidade mínima de retransmissões necessárias para se comunicar com todos os nós sensores da rede. A topologia híbrida é possível e pode ser uma interessante solução, dando

confiabilidade nas aplicações de maior criticidade e menor latência em aplicações de monitoramento.

Outra topologia clássica em RSSF é a *Mesh*, esta é caracterizada pela formação descentralizada de uma malha de comunicação, os nós sensores têm autonomia para troca informações com seus vizinhos e estabelecer seus links de comunicação. Esta arquitetura tem a vantagem de ser auto configurar e se adaptar as condições do momento mas exige alto processamento computacional nos nós sensores e dificulta e engessa o gerenciamento da rede. O foco deste trabalho são as arquiteturas de rede centralizadas. Na Figura 4 é ilustrado as topologias citadas.

Figura 4 - Topologias



Triângulos em verde com NS representam os nós sensores e em laranja com B, as bases.

2.5 Modulação FSK

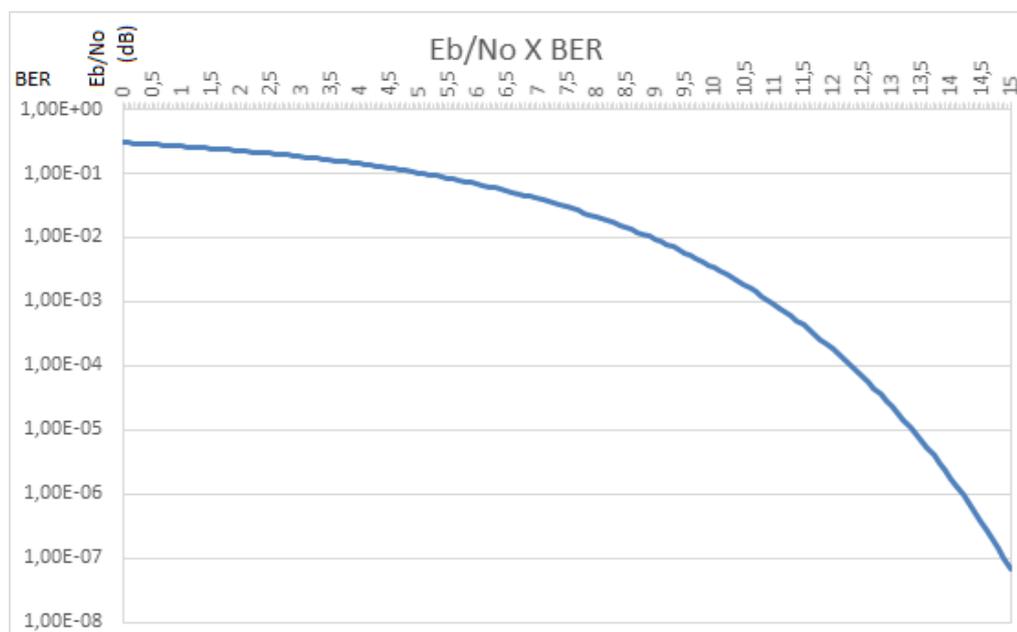
A modulação FSK (*Frequency-Shift Keying*) é bastante antiga e por algum tempo não tem sido utilizada em rede de sensores sem fio. Recentemente o padrão 802.15.4g[IEEE], na revisão de 2012, introduziu esta modulação com uma das possibilidades. Esta alternativa é interessante pela simplicidade, mas principalmente porque é possível variar banda ocupada do sinal modulado alterando o desvio de frequência, diferente de modulações como QPSK (*Quadrature Phase Shift Keying*) e ASK (*Amplitude Shift Keying*), onde a banda ocupada depende unicamente da taxa de comunicação (LATHI, 1998).

Encontra-se na literatura (Alkydiz, 2011) uma informação que é particularmente relevante para este trabalho, a relação E_b/N_0 com a BER (*Bit Error Rate*). E_b/N_0 trata-se da relação entre a energia do bit e o ruído térmico AWGN (*Additive White Gaussian Noise*). Na modulação FSK esta relação é dada pela equação abaixo.

$$BER = \frac{1}{2} \exp\left(-\frac{E_b}{2N_0}\right)$$

Esta relação implica no gráfico representado pela Figura 5.

Figura 5 – Gráfico E_b/N_0 X BER



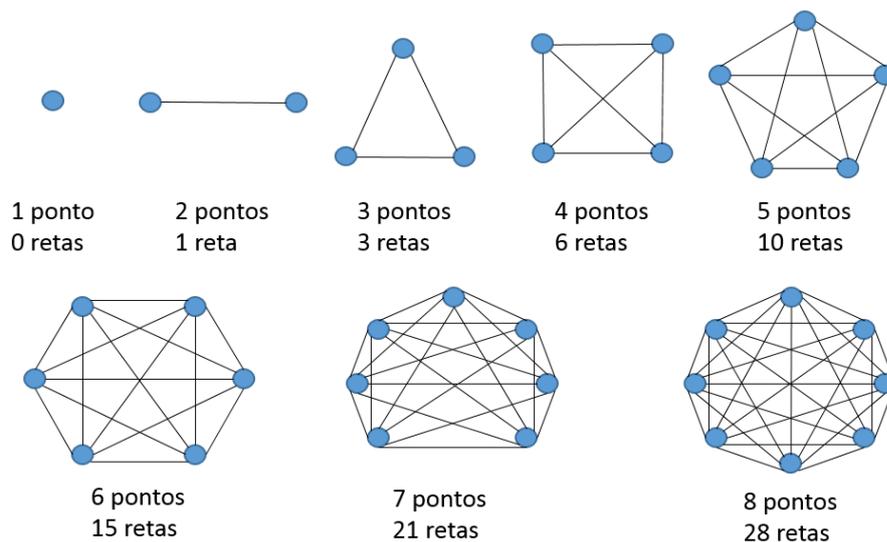
3 Análise de Tópicos em RSSF

Para embasar a proposta, foram desenvolvidas análises sobre o impacto do aumento do número de nós na rede, o paradigma para determinação de melhores rotas e a relação entre a RSSI e a BER. Estes tópicos são tratados na sequência.

3.1 Análise do Crescimento da Rede

A topologia física de uma RSSF é uma possível rede totalmente conectada. Esta situação ocorrerá caso todos nós estejam ao alcance de todos outros, este é o caso com maior quantidade de possíveis links de comunicação. Utilizando a geometria, pode-se considerar cada nó como um ponto e cada enlace como uma reta. A medida que aumenta-se o número de pontos em um determinado conjunto, o número de retas interligando todos estes ponto aumenta de forma crescente e não linear, uma progressão aritmética, como apresentado na Figura 6.

Figura 6 – Crescimento da Rede



Observando o comportamento do crescimento do número de retas a partir do crescimento do número de pontos, nota-se que a cada novo ponto adicionado, acrescenta-se também o número de retas com igual quantidade de pontos anterior. Com 1 ponto não existe reta. Inserindo um novo ponto, se faz necessário uma reta conectando-os. Inserindo outro ponto, somando 3 pontos, se faz necessário mais

duas retas que o conectem aos dois pontos anteriores, somando 3 retas. Inserindo o quarto ponto se faz necessário mais 3 retas que o conecte aos 3 pontos anteriores, somando 6 retas. Inserindo o quinto ponto se faz necessário mais 4 retas que o conecte aos 4 pontos anteriores, somando 10 retas. E assim sucessivamente.

Assim sendo, conclui-se que o número de retas será igual ao somatório do número de pontos de zero ao número de pontos menos um. A equação que descreve matematicamente esta relação entre o número de retas com o número de pontos é:

$$r = \sum_{i=1}^{p-1} i$$

Sendo r número de retas e p o número de pontos.

Uma outra abordagem que pode ser feita para se chegar ao total de retas em um conjunto determinado de pontos é a partir do tabelamento. Numerando-se os pontos de zero até seu número, é possível tabela-los de forma a representar as retas que os conectam binariamente, ou existe ou não existe reta. Para isso parte-se da premissa que não existe reta conectando um ponto com ele mesmo e que não importa a ordem dos pontos, se uma reta conecta 1 a 2, não existe reta conectando 2 a 1. A Tabela 2 representa uma matriz de conectividade indicando as retas existentes entre pontos nove pontos distintos.

Tabela 2 – Matriz da Existência de Retas

0	1	2	3	4	5	6	7	8	9
1	0	1	1	1	1	1	1	1	1
2	0	0	1	1	1	1	1	1	1
3	0	0	0	1	1	1	1	1	1
4	0	0	0	0	1	1	1	1	1
5	0	0	0	0	0	1	1	1	1
6	0	0	0	0	0	0	1	1	1
7	0	0	0	0	0	0	0	1	1
8	0	0	0	0	0	0	0	0	1
9	0	0	0	0	0	0	0	0	0

Esta é uma tabela com seu eixo diagonal zerado e com os valores inversamente espelhados em torno deste eixo, metade superior igual a 1 e inferior

igual a 0. Sendo assim, a quantidade de 1s nesta tabela será a metade da subtração do número de elementos total com a quantidade de elementos em seu eixo diagonal. O total de elementos será sempre o número de pontos elevado ao quadrado, pois trata-se da representação de uma matriz quadrada. A subtração entre o total de elementos com a quantidade destes no eixo diagonal, ou seja, um número elevado ao quadrado menos ele mesmo, pode ser interpretado como o próprio número vezes ele menos um, como representado na Equação X.

$$n^2 - n = n \times (n - 1)$$

Portanto também podemos representar matematicamente a relação entre o total de retas com total de pontos pela seguinte equação:

$$r = \frac{p \times (p - 1)}{2}$$

Sendo r número de retas e p o número de pontos. Neste último caso, 9 pontos e 36 retas, resultado que confere com a última tabela apresentada.

Tem-se, então, duas abordagens diferentes envolvendo as mesmas variáveis. Para que coexistam, ambas devem ser verdadeiras e necessariamente chegar ao mesmo resultado. Sendo que esta é uma forma de validá-las. A equação seguinte representa a condição necessária para que tais análises sejam verdadeiras.

$$\sum_{i=1}^{p-1} i = \frac{p \times (p - 1)}{2}$$

De fato, existe uma propriedade para somatórios em progressão aritmética que se enquadra perfeitamente nesta condição. Esta propriedade está representada pela equação:

$$\sum_{i=1}^n i = \frac{n \times (n + 1)}{2}$$

Substituindo n por p retorna-se na condição necessária, validando a condição e reforçando a validação da análise.

Em redes de sensores sem a análise é análoga, considerando-se que cada ponto pode ser um nó sensor as retas sejam os links de comunicação entre

eles. Existe a particularidade do link ser bidirecional, ou seja, faz diferença entre o nó 1 enviar um pacote para o nó 2 e o nó 2 enviar o pacote para o nó 1, é a mesma relação entre *uplink* e *downlink*. Neste caso cada reta representa dois links, fazendo com que o total de links possa ser representado como na seguinte equação.

$$l = n \times (n - 1)$$

Sendo l o número de links e n o número do total de nós sensores. Caso uma RSSF tenha dez nós, por exemplo, serão noventa possíveis links de comunicação entre eles, caso sejam vinte nós, serão trezentos e oitenta links possíveis. Nota-se que coletar as RSSIs de todos os possíveis links pode não ser uma tarefa trivial, sua complexidade pode variar dependendo da frequência de amostragem necessária, do tamanho da rede e da capacidade computacional do nó.

3.2 Determinação de Melhores Rotas

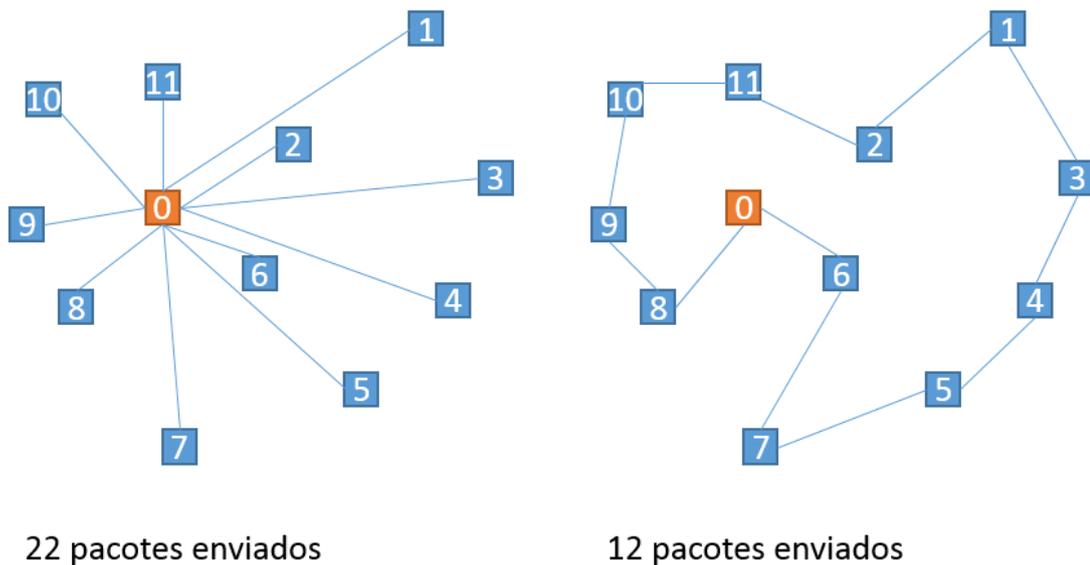
Para que se determine as melhores rotas é preciso conhecer as exigências das aplicações inseridas na rede, envolve frequência de requisição e suas criticidades, e a confiabilidade dos links de comunicação entre os nós. É preciso conhecer primeiro as possíveis topologias de rede e suas implicações para depois determinar melhores rotas.

3.2.1 Saltos X Topologia

Em redes de comunicação de dados a latência é caracterizada pelo tempo entre requisição de dados e a resposta entre os elementos que se comunicam. Esta latência é causada pelos tempos de transmissão, propagação e processamento do pacote. O resultado da soma destes três elementos é a latência total da comunicação entre dispositivos. Quando as retransmissões são necessárias, a latência deve considerar os tempos gastos desde a requisição até os dados de resposta processados. Portanto, diminuindo a quantidade total de transmissões de pacotes de uma rede, diminui-se também a latência total do sistema.

Para analisar estes fundamentos convém uma comparação entre a topologia em estrela e em anel pois ambas possuem número fixos de ligações para um dado número de nós. As topologias em árvore ou híbridas são de difícil comparação pois podem assumir inúmeras formas para cada quantidade de nós. Assim sendo, considerando, por exemplo, uma rede com onze nós sensores mais a base. Para a topologia seja em estrela, o total de pacotes transmitidos é de 22, enquanto que, na topologia em anel o total de pacotes transmitidos será de 12. Trata-se de 54,54% do montante anterior. NA Figura 7 é apresentado estes cenários.

Figura 7 – Envios de Pacotes por Topologia



Pode-se observar que caso as requisições sejam feitas nó por nó, o número total pacotes transmitidos, links de transmissão, será o duas vezes o número total de nós. Cada reta representa duas comunicações unidirecionais, *Uplink* e *Downlink*, ou seja, $l = 2 \times n$.

Em contrapartida, caso todas as requisições sejam feitas por uma única rota, o total de pacotes transmitidos, ou links criados, será o número total de nós sensores mais 1, isto é, $l = n + 1$. Sendo l o total de links e n o total de nós sensores.

Relacionando estas duas estratégias, com o total de nós tendendo ao infinito e calculando o percentual dentre os totais de links.

$$\lim_{n \rightarrow \infty} \frac{n + 1}{2 \times n} \times 100 = 50\%$$

Isso mostra que em redes suficientemente grandes, para se obter informações de todos os nós da rede utilizando a mesma rota para isso, o número de pacotes enviados cairá pela metade. De fato, os 55% ocorrem com n igual a 10 e os 51% com n igual a 50.

Esta análise demonstra como é possível minimizar consideravelmente a latência total da comunicação da rede adotando topologias que formem a malha da rede com menos saltos. Em contrapartida, existe um fator que torna praticamente inviável a criação de grandes rotas sem estratégias de contorno, a probabilidade de perda de pacotes (PER – *Packet Error Rate*). A chance do pacote se perder sempre existirá e se propagará por todo o caminho, tornando grandes rotas inviáveis em determinados casos.

3.2.2 PER X Saltos

A comunicação entre dispositivos em qualquer tipo de rede é sempre sujeita a falhas de transmissão e estas falhas podem comprometer o funcionamento do sistema como um todo. Por isso existem mecanismos para minimizar ou corrigir estas falhas. Tratando especificamente de RSSF, estes mecanismos não devem onerar os elementos da rede. De modo geral, nós sensores possuem baixa capacidade de processamento e possuem o propósito central de realizar medições ou atuar e controlar periféricos. Por isso, quanto menor for as atribuições de responsabilidade no controle da comunicação nos nós, mais dedicado a sua função principal o nó poderá ser.

O aproveitamento de rotas possui o indesejável efeito de aumentar a probabilidade de falha na comunicação. Este efeito pode ser calculado pelo produto das probabilidades de sucesso de cada link de comunicação pertencente a rota em questão.

$$P_s(R) = P_s(l1) \times P_s(l2) \times P_s(l3) \times \dots \times P_s(lT)$$

Sendo $P_s(R)$ a probabilidade de sucesso da rota e $P_s(l)$ a probabilidade de sucesso individual de cada link. O valor da PER será, $PER = 1 - P_s(R)$. Tomando como exemplo uma rede com onze nós sensores, como apresentado

anteriormente, e considerando que todos os links envolvidos em ambos os casos tenha uma PER de 3% (valor arbitrário apenas para realizar a análise), ou 97% de probabilidade de sucesso. Analisando os dois diferentes cenários, um com a requisição individual para cada nó e o outro com a rota de múltiplos destinos, temos que o primeiro é composto por onze rotas com apenas dois links em cada e o segundo apenas uma rota com treze links. Assim sendo, temos a seguinte probabilidade de sucesso para cada rota do primeiro cenário.

$$P_s(R) = 0,97 \times 0,97 = 0,9409 = 94,09\%$$

Esta probabilidade de sucesso resulta em uma PER de 5,91% para cada rota. Expandindo a análise, utilizando os mesmos princípios para a propagação do erro, para o conjunto das onze rotas temos que a probabilidade de sucesso de todas elas será:

$$P_s = 0,9409^{11} = 0,5116 = 51,16\%$$

Neste caso a PER não será o complemento da probabilidade de sucesso, continuará sendo de 5,91% pois a falha em um link não implica na perda total de dados da rede, penalizando individualmente cada nó sensor. O termo PER está sendo empregado pelo ponto de vista da gerencia, que é o elemento observador com capacidade para identificar que uma requisição efetuada não obteve resposta. Por isso a probabilidade de erro não varia com a mudança do número de rotas, mas sim com o número de saltos, ou retransmissões, que a rota possui.

No segundo caso, na qual todos os nós pertencem a mesma rota, a falha em qualquer um dos links implica na interrupção do fluxo de dados e a consequente perda de toda a informação. Assim sendo, a probabilidade de sucesso da rota será:

$$P_s(R) = 0,97^{12} = 0,6938 = 69,38\%$$

Esse resultado implica em uma PER de 30,62%, sendo este valor maior que a PER em cada rota do caso anterior, mas a probabilidade de sucesso, onde não há perda de nenhum pacote, é maior neste segundo que no primeiro.

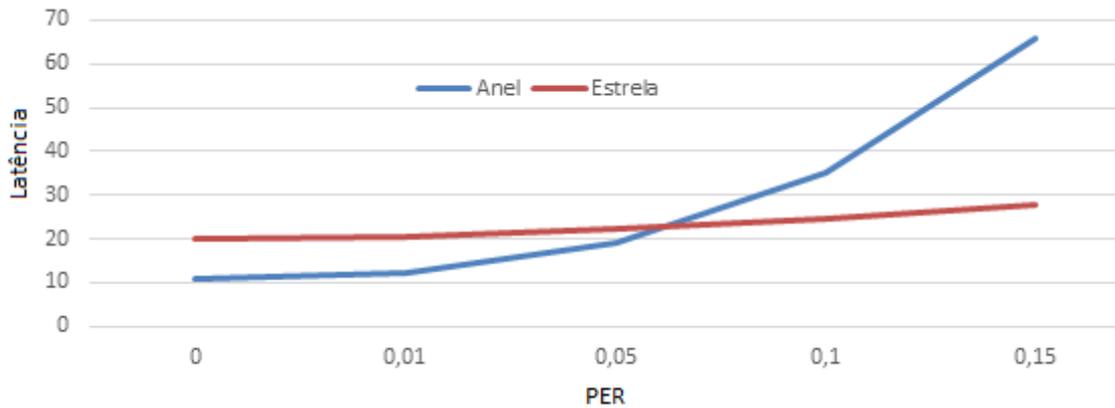
3.2.3 Latência X PER

As análises realizadas até aqui demonstraram que latência e PER se relacionam na atribuição das rotas. Para que se possa determinar as melhores rotas de forma eficiente é preciso equilibrar estes dois elementos de modo a atender as necessidades que foram atribuídas a rede. A perda de pacotes, mensurada pela PER, implica na necessidade de se retransmitir pacotes, ou seja, para se ter a frequência de aquisição de dados pré-estabelecida pela aplicação é necessário que se corrija o intervalo de aquisição de acordo com o valor da PER. Supondo que determinada aplicação precise de uma medição a cada 2 segundos e a comunicação implique em uma PER de 50%, o intervalo entre as requisições deve ser de 1 segundo. Isso demonstra que, na prática, a PER, assim como o número de saltos, impacta na latência.

Pode-se então aplicar uma métrica temporal que possibilite a comparação objetiva de diferentes rotas. Dividindo o número de saltos da rota pela probabilidade de sucesso da mesma, chega-se a esta métrica temporal que infere o tempo médio gasto para realizar a comunicação entre os nós em questão. Para transforma-la em segundos basta multiplicar este valor pelo tempo gasto em cada salto. Chega-se então no tempo médio gasto para concluir a rota.

Para mensurar a latência total da rede temos que comparar redes com diferentes número de nós e diferentes PERs. Novamente foram utilizadas as topologias em anel e estrela para a comparação da latência de duas formas. Na primeira com o total de nós fixo em 10 e PER variável, calculou-se a latência para ambas topologias. Na Figura 8 é apresentado o gráfico resultante desta análise.

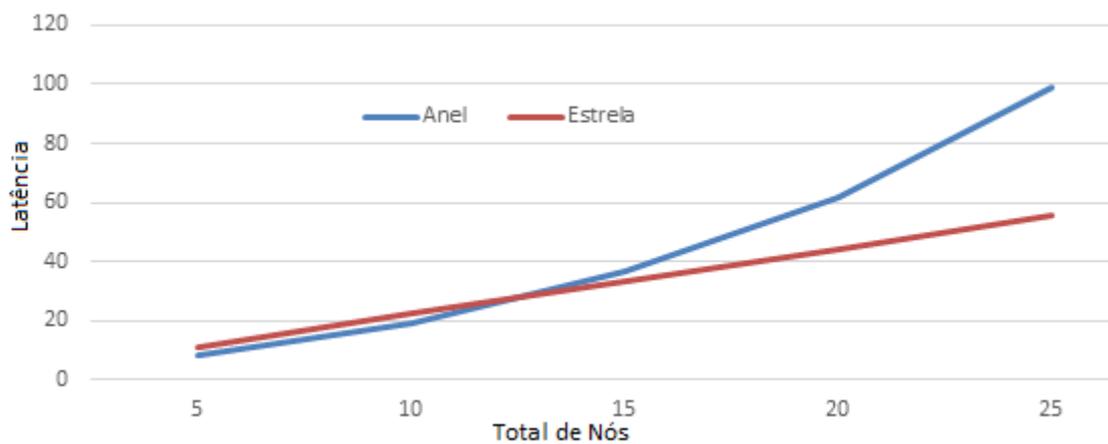
Figura 8 – Gráfico PER X Latência



Este gráfico mostra que, considerando uma RSSF com 10 nós sensores e que todos os enlaces tenham a mesma PER, a latência é menor na topologia em anel quando se tem uma PER de aproximadamente 5%. Essa diferença aumenta conforme aumenta-se a PER, demonstrando que a topologia em estrela é menos sensível a ela.

A segunda forma é fixando a PER, 5% neste caso, variando o total de nós e calculando as latências para ambas topologias como no exemplo anterior. Extrair assim o gráfico apresentado na Figura 9.

Figura 9 – Gráfico Total de Nós X Latência



O gráfico demonstra que em uma rede que tenha uma PER de 5% entre todos os nós, até um total de aproximadamente 12 nós é vantagem usar a topologia em anel e acima disso, estrela é mais eficaz.

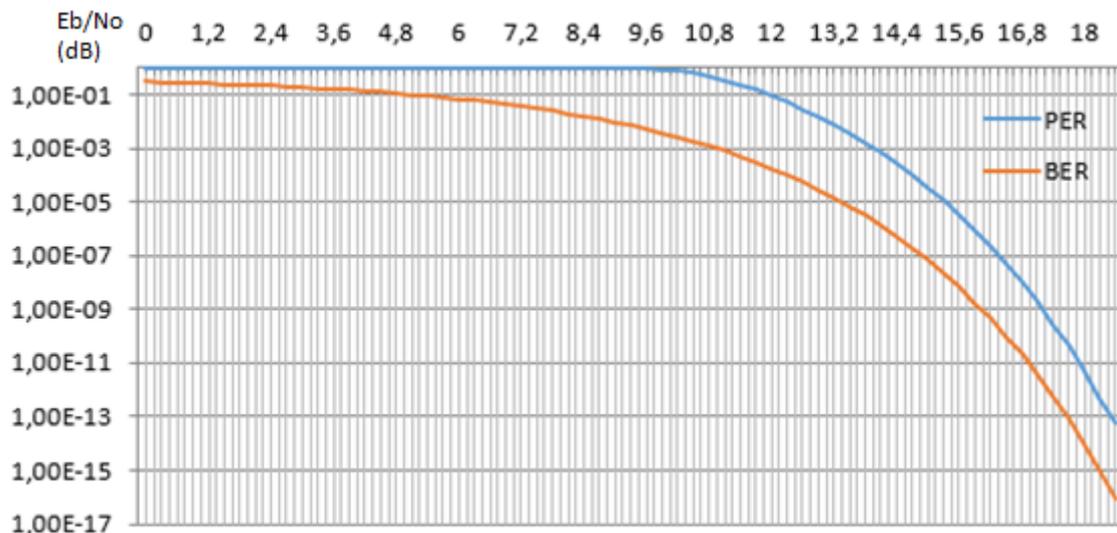
3.3 Estimativa da PER pela RSSI

O Capítulo 2.5, Modulação FSK, apresenta a relação entre a BER e E_b/N_0 para esta modulação. Conforme a teoria apresentada, é possível estimar a PER a partir da BER. Estas duas grandezas se relacionam pela seguinte expressão:

$$PER = 1 - (1 - BER)^N$$

Sendo N o número de bits do pacote. Neste trabalho a rede desenvolvida opera com pacotes de 512 bits. A Figura 10 apresenta um gráfico relacionando as curvas de PER e BER por E_b/N_0 .

Figura 10 – Gráfico E_b/N_0 X Per e BER



A curva da PER, em azul, tem particular relevância neste trabalho. O fato desta estar relacionada com E_b/N_0 e não com a RSSI inviabiliza sua utilização diretamente na gerência da rede. É necessário que se converta E_b/N_0 para RSSI.

Em teórica, para realizar a conversão de E_b/N_0 para RSSI é preciso fazer algumas estimativas e considerações. Esta abordagem pode não representar muito bem a realidade, por isso também foram realizados testes experimentais para confrontar a teoria com a prática.

3.3.1 Estimativa Teórica

E_b/N_o não é uma métrica prática para se utilizar na análise de desempenho pelo gerente da rede. E_b/N_o trata-se de uma razão entre a energia do bit e a densidade espectral de ruído, ambos medidos em Joules (J). E_b não é medido na prática, mas pode ser encontrado pela razão da potência média do sinal (W) pela taxa de transmissão ($1/s$), que é um parâmetro da configuração da rede.

O termo N_o não pode ser calculado, mas pode ser estimado pelas características do sistema, como o ruído térmico. O produto entre a densidade espectral de ruído (J) e sua banda ocupada ($Hz = 1/s$) implica na potência de ruído (W). Respeitando as definições das grandezas encontra-se a equação da relação sinal ruído (SNR), que é a razão entre a potência de sinal recebido e o ruído do sistema,

$$\frac{C}{N} = \frac{E_b}{N_o} \times \frac{f_b}{B_w},$$

OU

$$\frac{E_b}{N_o} = \frac{C}{N} \times \frac{B_w}{f_b}.$$

Sendo C a potência de recepção, N é o ruído composto pelo ruído térmico e a figura de ruído (ruído inerente do receptor causado pelo processo de amplificação e processamento do sinal), B_w é a banda ocupada e f_b é a taxa de transmissão. Neste ponto é interessante recordar que a banda ocupada da modulação FSK pode ser alterada com o desvio de frequência, obtendo-se bandas ocupada bem maiores que as modulações QPSK e ASK, por exemplo. Interessante observar que para certificação de rádios com esta modulação a banda ocupada deve ser maior de 500 kHz, como estabelecido pela resolução 506 da ANATEL [ANATEL].

Para igualar com a função da BER que utiliza o termo $E_b/2N_o$

$$\frac{E_b}{2N_o} = \frac{1}{2} \times \frac{C}{N} \times \frac{B_w}{f_b}$$

O ruído térmico pode ser calculado pela equação de Boltzmann representada abaixo.

$$N1 = kTBw$$

Sendo N1 a potência do ruído térmico, k é a constante de Boltzmann (1.380650×10^{-23} J/K) e T é a temperatura efetiva em Kelvin.

Inserindo a potência do ruído térmico na equação anterior temos:

$$\frac{Eb}{2No} = \frac{1}{2} \times \frac{C}{kTBw \times N2} \times \frac{Bw}{fb}$$

$$\frac{Eb}{2No} = \frac{1}{2} \times \frac{C}{kT \times N2} \times \frac{1}{fb}$$

$$\frac{Eb}{2No} = \frac{C}{2kTN2fb}$$

Sendo N2 a figura do ruído. Utilizando esta equação para a aplicação deste trabalho é preciso deixá-la em função apenas da potência do sinal. Padronizando as outras grandezas temos os seguintes valores:

- T = 295K
- N2 = 10x
- fb = 4,8kbps

Temos então a seguinte equação

$$\frac{Eb}{2No} = \frac{C}{2 \times 1,38065 \times 10^{-23} \times 295 \times 10 \times 4,8 \times 10^3}$$

$$\frac{Eb}{2No} = \frac{C}{3,91 \times 10^{-16}}$$

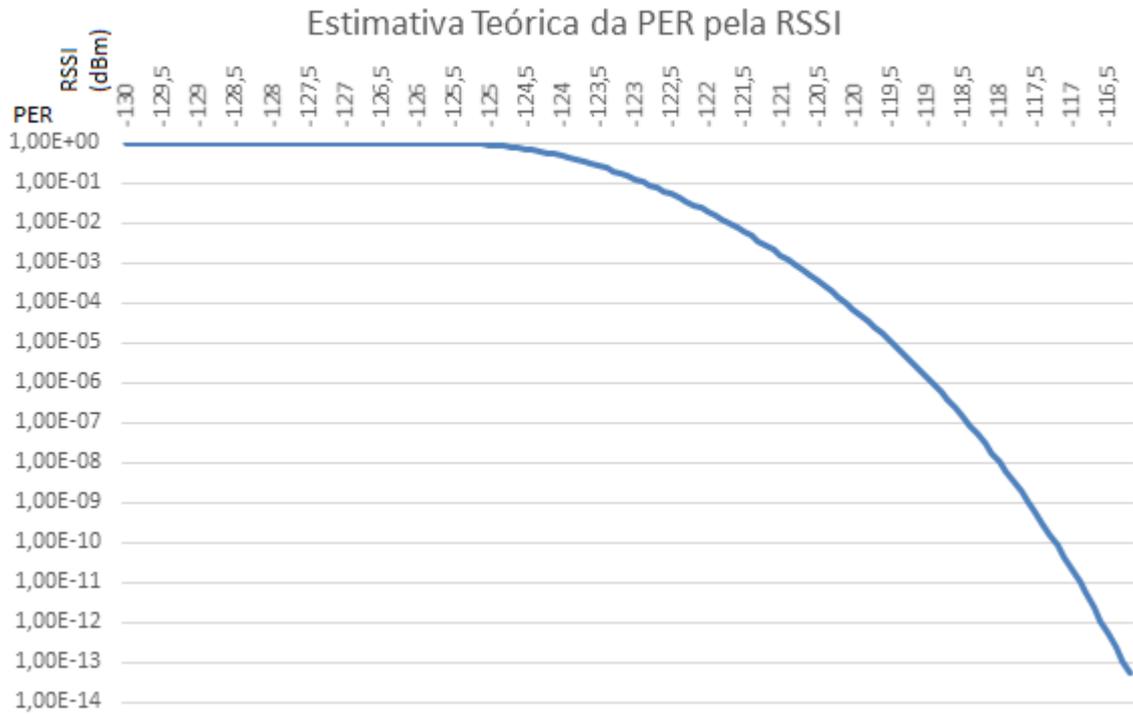
Eb/No é medido em dB, portanto é preciso adicionar a conversão de escala.

$$\frac{Eb}{2No} = C - 10 \log_{10}(1000 \times 3,91 \times 10^{-16})$$

$$\frac{Eb}{2No} = C + 134,08$$

Fazendo a conversão da RSSI para Eb/No e considerando a curva da estimativa da PER apresentada no capítulo anterior, obtém-se a estimativa teórica da PER pela RSSI. A Figura 11 apresenta um gráfico que ilustra esta relação.

Figura 11 – Gráfico RSSI X Estimativa Teórica da PER



4 Proposta

A proposta deste trabalho se divide em três etapas. Começando por uma estimativa da PER pela RSSI seguida por um protocolo para a coleta de RSSIs e finalizando com uma nova metodologia para a configuração de melhores rotas.

4.1 Estimativa da PER em Função da RSSI

Para obter estimativas de PER mais realista foram propostos experimentos práticos para verificar a perda de pacote. O propósito destes experimentos é de criar um cenário em que se possa transmitir uma grande quantidade de pacotes de modo a se obter uma potência de recepção com valor fixo, ou o mais próximo disso possível. Realizando a contagem de pacotes perdidos em diferentes intensidades de sinal é possível plotar uma curva que represente a estimativa de percas devido a RSSIs.

Para obter uma potência de recepção estável foi utilizada uma câmara blindada em um sistema cabeado, isolando o transmissor do receptor e evitando assim vazamentos de radiação e o *by-pass* nos atenuadores. A potência mínima de transmissão para que o transceptor do modulo utilizado opere é de -30dBm, sendo necessário a utilização de atenuadores para se obter a potência de recepção desejada. Estes elementos possuem diferentes valores de atenuação, 2, 3, 6, 10, 20 e 30dB, podendo ser combinados para se obter diferentes resultantes.

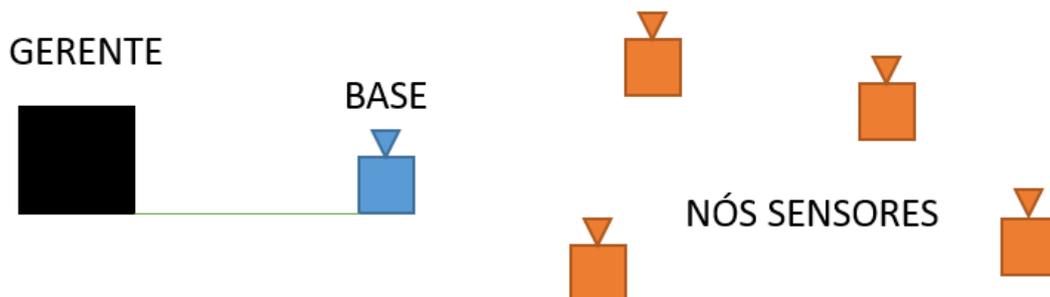
4.2 Protocolo Para Coleta de Dados

Para realizar os experimentos propostos foi desenvolvido um protocolo para RSSF que possibilita a coleta da RSSI entre todos os nós da rede. O princípio de seu funcionamento é puramente matemático, sendo desnecessário o armazenamento de qualquer tipo de tabela ou informação durante sua operação. O protocolo desenvolvido permite que todos os enlaces entre os nós sensores sejam percorridos de forma igualitária. Os pacotes são disparados pela base e retransmitidos pelos nós sensores até seu retorno a base. O pacote disparado

percorre uma rota que será determinada pelo endereço do primeiro nó destino (aquele que foi enviado pela base) e pelo número total de sensores da rede.

O protocolo desenvolvido opera de forma centralizada, determinística, com baixa exigência de recursos computacionais e adaptável a qualquer quantidade de nós. A rede é composta por três elementos diferentes, sendo eles, o *software* gerente, a base e os nós sensores, Figura 12. O software gerente está localizado no computador onde a base está conectada. A base é um elemento que opera de forma transparente, sem conversão de dados, enviando para a interface aérea toda informação que recebe via serial e enviando pela serial toda informação endereçada a ela vinda pela interface aérea. Os nós sensores são os elementos responsável pela identificação de destino do pacote, medições de RSSI, alocação destas informações no pacote e sua retransmissão.

Figura 12 – Gerente na RSSF



Este protocolo pode ser considerado centralizado, pois, somente o elemento gerente tem autonomia para requisição de dados sendo que estes ficam armazenado exclusivamente no próprio gerente. Os outros elementos participam passivamente do processo, somente reagindo aos pacotes que forem endereçados a eles.

É determinístico, pois o conjunto de rotas que cobrem todos os links entre todos os nós sensores é formado puramente por operações matemáticas básicas. Estes cálculos são realizados de duas formas utilizando apenas três parâmetros: o endereço do primeiro nó sensor da rota, o número de saltos realizados pelo pacote (este valor é incrementado a cada retransmissão) e o número total de sensores na

rede. Operações matemáticas com estes três parâmetros permitem calcular qual o endereço do próximo nó sensor de destino do pacote, podendo calcular a sequência de endereços de qualquer uma das rotas que formam o conjunto.

A baixa exigência de recursos computacionais se deve a baixa complexidade dos cálculos para endereçamento e a inexistência de qualquer tipo de tabela de roteamento, dispensando a necessidade de memória para tal.

A adaptação da rede para qualquer quantidade de nós não exige grandes mudanças. A informação do total de sensores na rede é uma necessidade do software gerente para que este realize os disparos de todo o conjunto de rotas. Esta informação é encaminhada no pacote sendo utilizada pelos nós sensores no cálculo de retransmissão do pacote, sendo que este dado não fica armazenado no nó, apenas retransmitida no pacote. A quantidade total de rotas e o número total de saltos são dependentes do número total de sensores, isso garante que as características da rede se adaptem de forma que todos os links sejam percorridos. Adaptações são necessárias caso o tamanho do pacote não seja suficiente para alocar todas as medidas de RSSI e LQI da rota, pois, quanto maior a rede, maior o número de saltos por rota e conseqüentemente maior é o tamanho mínimo necessário do pacote.

Existe uma peculiaridade com relação ao total de sensores ser par ou ímpar. Quando trata-se de um total ímpar, existe a simetria da rota, fazendo com que o pacote percorra os mesmos nós na descida e na subida. Caso o total seja par essa simetria não existe, a rota possui um caminho na qual não existe relação entre a descida e a subida do pacote.

Em uma RSSF com oito nós sensores (mais a base), como já demonstrado anteriormente, possui um total de 72 (setenta e dois) possíveis links de comunicação. O conjunto de rotas que representam as sequências de nós sensores que são caminhos para o pacote podem ser descritas com uma tabela de oito linhas por dez colunas, Tabela 3. Uma consideração importante a se fazer é com relação a base (0), todas as rotas devem, necessariamente, iniciar e terminar nela. Apesar de contar com 80 (oitenta) endereços, esta tabela representa apenas 72 (setenta e dois) enlaces.

Tabela 3 – Rotas do Protocolo de Coleta, 8 nós

	Saltos	1	2	3	4	5	6	7	8	9
Rota	Endereços									
1	0	1	2	4	7	3	8	6	5	0
2	0	2	3	5	8	4	1	7	6	0
3	0	3	4	6	1	5	2	8	7	0
4	0	4	5	7	2	6	3	1	8	0
5	0	5	6	8	3	7	4	2	1	0
6	0	6	7	1	4	8	5	3	2	0
7	0	7	8	2	5	1	6	4	3	0
8	0	8	1	3	6	2	7	5	4	0

Todos os possíveis enlaces de comunicação em uma RSSF com oito nós sensores estão descritos no conjunto das oito rotas da tabela acima. Em alguma das rotas está a comunicação de qualquer nó sensor com qualquer outro. Por exemplo, 1 se comunica com 6 na sétima rota durante o quinto salto, mas 6 se comunica com 1 na terceira rota durante o terceiro salto. Isso mostra que a rota não é simétrica, ou seja, não é possível caracterizar o momento da rota onde o pacote desce e onde sobe. Como dito anteriormente, esta característica se deve ao número total de sensores ser par.

Em uma RSSF com sete nós sensores (mais a base) é possível descrever todos os possíveis links de comunicação em uma tabela com sete linhas e nove, Tabela 4, totalizando 63 (sessenta e três) endereços e os 56 (cinquenta e seis) possíveis links de comunicação.

Tabela 4 - Rotas do Protocolo de Coleta, 7 nós

	Saltos	1	2	3	4	5	6	7	8
Rota	Endereços								
1	0	1	2	4	7	4	2	1	0
2	0	2	3	5	1	5	3	2	0
3	0	3	4	6	2	6	4	3	0
4	0	4	5	7	3	7	5	4	0
5	0	5	6	1	4	1	6	5	0
6	0	6	7	2	5	2	7	6	0
7	0	7	1	3	6	3	1	7	0

Neste caso nota-se a existência de simetria na rota, por exemplo, na primeira rota o pacote parte da base (0), percorre a sequência 1, 2, 4, 7, 4, 2, 1 e retorna para a base. É possível definir qual a descida e qual a subida do pacote, ou então quando ele está indo e quando está voltando. Nesta rota o nó sensor 7 é o limiar para esta caracterização, até nele o pacote desce e a partir dele o pacote sobe. Outra característica para este tipo de rota é que a comunicação bidirecional entre dois nós sensores quaisquer estão na mesma rota, por exemplo, 3 se comunica com 5 no segundo salto da segunda rota e 5 se comunica com 3 no quinto salto da mesma rota.

As particularidades entre a presença e ausência de simetria das rotas, inerentes do protocolo para a coleta de dados, podem ser relevantes dependendo do propósito atribuído ao protocolo. Estas particularidade não possuem grande relevância para o objetivo deste trabalho. Pode-se considerar que ambas as abordagens apresentam o mesmo desempenho no que se refere ao fato de ambas cobrir todos os links de forma igualitária, sem que algum link seja mais ou menos utilizado que o outro. Um ponto importante a se destacar é o fato do protocolo desenvolvido cumprir os objetivos para qual foi criado utilizando o mínimo de rotas possíveis. Esta afirmação se baseia no fato de que todas as rotas iniciam e terminam na base, isso faz com que seja necessário que a base envie diretamente um pacote para cada um dos outros nós, fazendo necessário que existam pelo menos um número rotas igual ao total de sensores. Obrigatoriamente não podem existir duas ou mais rotas que tenham os mesmos primeiros ou os mesmos últimos nós. O aumento do número de rotas diminuiria a necessidade de tamanho de pacote, com mais rotas, seriam necessários menos saltos por rota e, conseqüentemente, menos informação ao final da rota. Para aumentar o número de rotas e manter a divisão igualitária entre os links percorridos seria necessário que o número de rotas seja um múltiplo do total de sensores, caso contrário algum link, automaticamente, seria percorrido mais que outros. Esta necessidade não foi identificada neste trabalho e as rotas foram formadas da forma mais simples.

Como discutido anteriormente, as rotas são configuradas baseadas em operações matemáticas, e não por tabelas previamente estabelecidas. As tabelas com as rotas mostradas até então podem ser calculadas de duas maneiras diferentes, calculando o próximo nó sensor a partir do endereço de envio ou a partir

do primeiro nó sensor da rota. Os dois modos de realizar os cálculos se baseiam em operações matemáticas que envolvem o número de saltos com o endereço e ter nós sensores com endereços sequenciais, inteiros e começando em 1.

4.2.1 Destino do Pacote Determinado pelo Nó Sensor de Envio

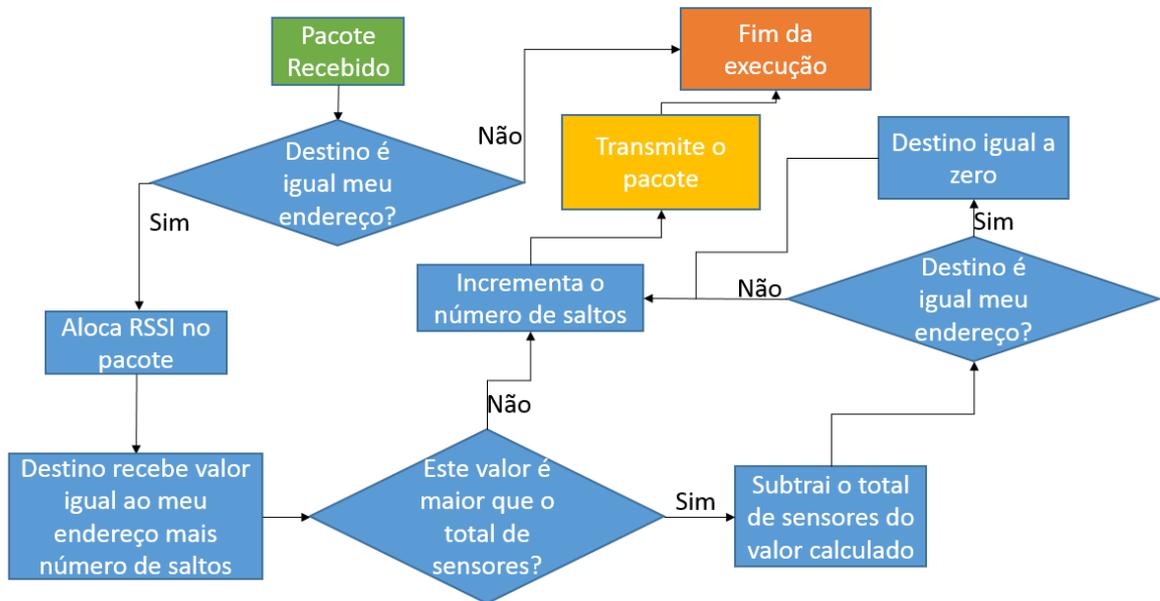
As etapas dos cálculos demonstrados aqui podem ser aplicados as tabelas que já foram apresentadas, começando pelo cálculo a partir do sensor de envio. Para exemplificar temos uma tabela com as rotas de uma RSSF com nove nós sensores, Tabela 5. O primeiro nó sensor da rota é definido pelo software gerente, a partir dele podemos calcular todo o restante sabendo o total de sensores da rede e o número de saltos realizados. Neste caso as três informações fundamentais que o pacote precisa carregar são o endereço do nó destino, número de saltos e total de sensores na rede.

Tabela 5 - Rotas do Protocolo de Coleta, 9 nós

	Saltos	1	2	3	4	5	6	7	8	9	10
Rota	Endereços										
1	0	1	2	4	7	2	7	4	2	1	0
2	0	2	3	5	8	3	8	5	3	2	0
3	0	3	4	6	9	4	9	6	4	3	0
4	0	4	5	7	1	5	1	7	5	4	0
5	0	5	6	8	2	6	2	8	6	5	0
6	0	6	7	9	3	7	3	9	7	6	0
7	0	7	8	1	4	8	4	1	8	7	0
8	0	8	9	2	5	9	5	2	9	8	0
9	0	9	1	3	6	1	6	3	1	9	0

Nesta primeira forma de cálculo apresentada, os endereços devem, necessariamente, ser calculados sequencialmente em linha. O próximo endereço será a soma do número de saltos ao endereço atual, caso o destino tenha um valor maior que o total, subtrai-se o valor do total de nós do destino, caso o endereço de destino seja igual ao endereço atual o destino é a base (0). Representando o laço de execução do nó sensor por esta abordagem temos o fluxograma apresentado na Figura 13.

Figura 13 – Fluxograma do Protocolo de Coleta para os Nós Sensores



Este fluxograma representa todos os laços de operação dos nós ao receberem um pacote. Nota-se que em momento algum existe o armazenamento de dados (com exceção do endereço do nó que é prestabelecido), apenas verificação de endereçamento, alocação dos valores medidos no pacote e o cálculo do endereço do próximo nó sensor da rota. Os laços de execução da retransmissão podem ser descritos na forma de pseudocódigo, Figura 14. Os valores do nó destino do pacote, número de saltos, total de nós e endereço do nó estão representados como destino, saltos, total_nós e meu_endereço respectivamente. A função responsável pela transmissão do pacote está representada como transmite.pacote.

Figura 14 – Pseudocódigo do Protocolo de Coleta para os Nós Sensores

Pseudocódigo

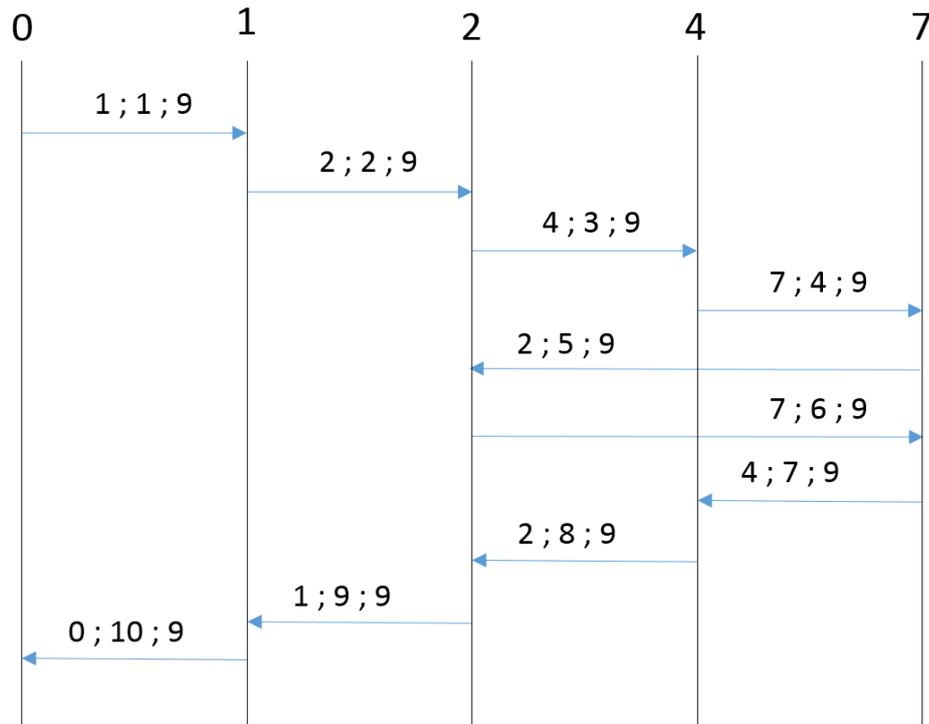
```

leia (destino)
leia (saltos)
leia (total_nós)
se (destino = meu_endereço) então
    destino = meu_endereço + saltos
    se(destino > total_nós) então
        destino = destino – total_nós
        se (destino = meu_endereço) então
            destino = 0
        fim-se
    fim-se
fim-se
saltos = saltos + 1
transmite.pacote
fim-se

```

Este pseudocódigo representa a forma na qual os valores presentes e necessários para a formação da rota interagem. Três dos quatro valores estão alocados no pacote (endereço de destino, número de saltos e total de sensores), o outro fica armazenado no nó (o endereço único do nó na rede), sendo esta a única informação que o nó tem. A execução sucessiva destes laços pelos nós sensores resultam na retransmissão sucessiva do pacote e a consequente criação de rotas. Analisando a formação destas rotas pelo diagrama temporalmente percebe-se como os valores necessários para a retransmissão se comporta. Como exemplo, o diagrama abaixo representa a retransmissão e os valores do pacote necessários para o roteamento da primeira rota da Tabela 5 (com nove nós sensores na rede), Figura 15. Temos o endereço de destino no primeiro campo, o número de saltos no segundo e o total de sensores no terceiro. As linhas verticais representam os nós sensores e seus respectivos endereços acima delas.

Figura 15 – Diagrama Temporal do Protocolo de Coleta

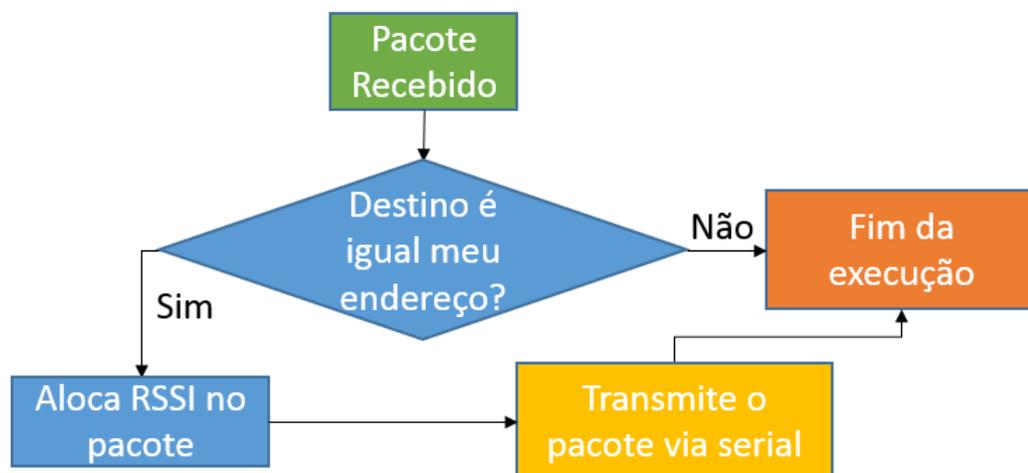


A rota em questão trata-se da sequência 0, 1, 2, 4, 7, 2, 7, 4, 2, 1 e 0 de endereços. Nesta rota o nó sensor 2 é mais utilizado que os demais, porém este efeito ocorre em todas as rotas com nós diferentes, fazendo com que nenhum nó seja mais ou menos utilizados que outros ao percorrer todas as rotas. Nota-se que o segundo campo, saltos, é incrementado em 1 a cada retransmissão. O terceiro campo, total de sensores, é constante e definido pelo gerente. O primeiro campo, endereço destino, é calculado de acordo com método aqui demonstrado.

O software gerente dispara o pacote a partir da base. O 1 recebe o pacote e soma seu endereço ao número de saltos ($1+1=2$) e retransmite o pacote com destino e número de saltos atualizados. 2 recebe o pacote e soma 2 ao seu endereço ($2+2=4$) e retransmite o pacote para o destino 4. Por sua vez, 4 recebe o pacote, soma o número de saltos ao seu endereço e retransmite o pacote para 7 ($4+3=7$). Em 7 um novo laço de execução é utilizado, somar os saltos ao seu endereço resulta em 11 ($7+4=11$), porém este valor é maior que o total de sensores, então subtrai-se o resultado do total de sensores ($11-9=2$), retransmite-se o pacote. Em 2 é somado o número de saltos ao endereço ($2+5=7$) e o pacote é retransmitido. O 7 recebe o pacote novamente mas desta vez com 6 saltos ($7+6=13$), como o resultado é maior que o total de sensores o laço de subtração é executado ($13-9=4$)

e o pacote retransmitido. Em 4 com 7 saltos ($4+7=11$) o laço de subtração é acionado ($11-9=2$) e o pacote é retransmitido. 2 recebe o pacote novamente e executa o laço da soma, da subtração ($2+8-9=1$) e retransmite o pacote. Em 1 é executado o laço da soma e da subtração, mas como os dois valores são iguais (saltos igual a 9 e total de sensores também igual a 9) fica evidente que o destino resultará no seu próprio endereço ($1+9-9=1$), neste caso o laço de envio para a base é acionado ao identificar que o destino é o próprio nó, então destino recebe 0 e o pacote é retransmitido. Todos os nós que retransmitiram também incrementaram o número de saltos e alocaram os dados pertinentes no pacote. Ao chegar na base o pacote é enviado via serial ao software gerente com as informações de todos os nós da rota. O processo executado pela base é mais simples que dos nós pois ela não precisa calcular o destino do pacote, apenas alocar a RSSI e retransmiti-lo ao gerente. A Figura 16 representa o fluxograma que descreve este processo.

Figura 16 – Fluxograma do Protocolo de Coleta para a Base



4.2.2 Destino do Pacote Calculado na Recepção

Uma segunda abordagem para efetuar os cálculos de retransmissão do pacote e conseqüente formação da rota é a partir do primeiro nó sensor da rota. Existe uma diferença conceitual entre as duas abordagens. Enquanto na primeira o endereço de destino era enviado no pacote, nesta segunda o pacote não tem o campo de destino, apenas o primeiro nó sensor da rota, o total de nós na rede e a

quantidade de saltos. Sabendo estas três informações o nó sensor calcula se o pacote é para ele ou não. É como enviar uma tabela e as coordenadas de localização com apenas três valores. O total de nós sensores é suficiente para montar toda a tabela, o endereço do primeiro nó equivale a linha, ou rota, e o número de saltos equivale a coluna da tabela. Na prática esta tabela não é construída pelo nó sensor, apenas é feito o cálculo para verificar se o destino do pacote é o próprio nó ou não.

Ao receber o pacote, o nó calcula seu endereço de destino, caso este endereço seja o do próprio nó o pacote é processado e retransmitido, caso contrário ele é descartado. A lógica que envolve o cálculo de destino do pacote parte do mesmo princípio da forma de cálculo apresentada anteriormente. A cada salto o nó soma o total de saltos efetuados até o momento e soma este valor ao seu endereço. Neste caso será necessário calcular o soma do total de saltos realizados em cada momento em que o pacote é retransmitido. Por exemplo, caso pacote tenha 5 saltos, na retransmissão anterior ele tinha 4, e na anterior,3, e assim sucessivamente. Neste momento a soma do total de saltos realizados em cada retransmissão será 15 (5+4+3+2+1). Esta situação é análoga a do capítulo 3.1 (Análise de Crescimento da Rede), para se chegar ao total de possíveis links de uma rede é preciso realizar uma operação de progressão aritmética de incremento de 1, assim como neste caso. Assim sendo, como demonstrado anteriormente, podemos realizar esta operação com a seguinte operação.

$$T = \frac{s \times (s - 1)}{2}$$

Sendo T a soma dos saltos em cada nó sensor e s o número de saltos do pacote.

Ao obter este valor, soma-se a ele o valor do endereço do primeiro nó sensor da rota. Temos então a seguinte equação.

$$D = P + T$$

Sendo D o endereço do destino do pacote e P o endereço do primeiro nó sensor da rota.

Assim como na forma de cálculo demonstrada anteriormente, neste caso também se faz necessária a delimitação do grupo numérico de trabalho. Esta delimitação é feita pela comparação do endereço de destino com o total de nós sensores da rede. Caso o endereço de destino tenha um valor maior que o total de nós da rede, subtrai-se o total de nós do endereço de destino. O pseudocódigo que representa o processo dos nós para este caso está apresentado pela Figura 17.

Figura 17 – Pseudocódigo do Protocolo de Coleta 2

Pseudocódigo

```

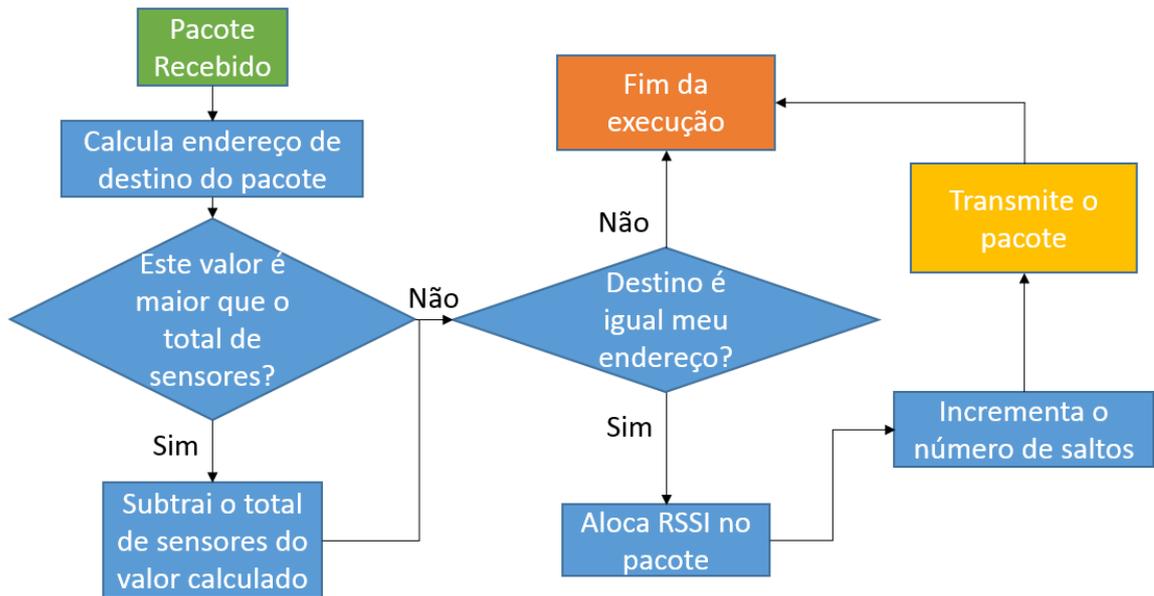
leia (nó_1)
leia (saltos)
leia (total_nós)
destino = nó_1 + saltos x (saltos-1) / 2
se(destino > total_nós) então
    destino = destino – total_nós
fim-se
se (destino = meu_endereço) então
    saltos = saltos + 1
fim-se
transmite.pacote
fim-se

```

Sendo nó_1 o endereço do primeiro nó da rota, saltos o número de saltos do pacote, total_nós o total de nós sensores da rede, destino o endereço na qual o pacote é destinado, meu_endereço é o endereço do próprio nó sensor e transmite.pacote a função de transmissão do pacote.

A Figura 18 representa o diagrama em blocos deste mesmo processo.

Figura 18 – Fluxograma do Protocolo de Coleta para os Nós Sensores 2



Para exemplificar este modo de operação, tem-se uma rede com um total de 5 nós sensores. As rotas que correspondem a esta rede está representada na Tabela 6.

Tabela 6 – Rotas do Protocolo de Coleta, 5 Nós

	Saltos	1	2	3	4	5	6
Rota	Endereços						
1	0	1	2	4	2	1	0
2	0	2	3	5	3	2	0
3	0	3	4	1	4	3	0
4	0	4	5	2	5	4	0
5	0	5	1	3	1	5	0

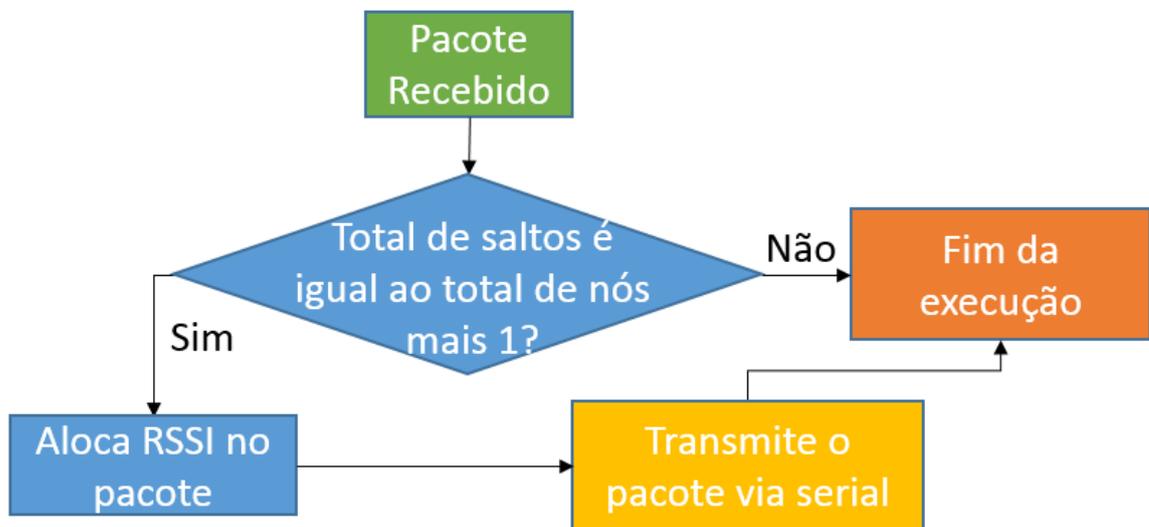
Em destaque, na terceira rota e quarto salto o valor 4. Para se encontrar este valor calcula-se o endereço de destino com a seguinte equação.

$$D = 3 + \frac{4 \times (4 - 1)}{2}$$

Neste caso, o destino encontrado é igual a 9, que é maior que o total de nós(5). Então, subtrai-se 5 de 9, chegando ao valor 4, que corresponde ao endereço de destino do pacote. Pode-se encontrar qualquer endereço da tabela partindo deste mesmo princípio com exceção do 0.

Fazendo esta mesma operação para o sexto salto da primeira coluna, por exemplo, encontra-se o valor 1, que é o endereço do nó sensor que enviou o pacote. Isso ocorre sempre no último salto de qualquer rota para qualquer quantidade de nós, repete-se o endereço do último envio. Então, nenhum nó sensor processará o pacote, já que quem envia não lê seu próprio pacote. Por isso faz-se necessário que a base tenha um algoritmo que entenda este momento. Para isso basta confrontar o número de saltos com o total de sensores da rede, caso o número de saltos igual ao total de saltos mais 1, este pacote é para a base. A Figura 19 apresenta o fluxograma que representa o processo da base.

Figura 19 – Fluxograma do Protocolo de Coleta para a Base



Este processo efetuado pela base também funcionaria no método demonstrado no capítulo anterior, porém exige uma operação matemática a mais para se executar. Esta operação se repetirá mesmo que o pacote não seja para a base.

As rotas criadas por meio desta abordagem são as mesmas criadas pela abordagem apresentada no capítulo anterior, A diferença se dá apenas pelo método utilizado para criá-las.

4.2.3 Atribuição de Rotas Alternativas

Devido as possíveis falhas na comunicação, algumas rotas podem ter uma chance de sucesso muito maiores que as outras. Por isso criar formas que permitam determinar rotas alternativas, utilizando métodos semelhantes aos apresentados, é uma maneira de manter as características da rede com uma minimização de eventuais danos causados por rotas pouco confiáveis.

Uma forma de criar estas rotas com os mesmos princípios dos métodos anteriores é a inversão da lógica de cálculos do endereço de destino. Ao invés de somar o número de saltos, subtrair. Esta inversão torna necessário a mudança da correção do grupo numérico de trabalho. Antes o endereço de destino poderia ser maior que o total de nós, agora este endereço pode ser menor ou igual a zero. De forma anômala ao anterior, caso isso aconteça, faz-se necessário somar o total de nós ao endereço do sensor.

Tomando como exemplo uma rede com 5 nós sensores, como no último exemplo. As rotas formadas por este método estão apresentadas na Tabela 7.

Tabela 7 – Rotas Alternativas do Protocolo de Coleta, 5 Nós

	Saltos	1	2	3	4	5	6
Rota	Endereços						
1	0	1	5	3	5	1	0
2	0	2	1	4	1	2	0
3	0	3	2	5	2	3	0
4	0	4	3	1	3	4	0
5	0	5	4	2	4	5	0

A utilização dos dois métodos de formação de rotas paralelamente tende a homogeneizar mais a coleta de RSSI entre os nós sensores. A criação de rotas alternativas também pode ser utilizada efetuando os cálculos a partir do nó de envio ou na recepção.

4.3 Determinação de Melhores Rotas

A proposta para determinação de melhores rotas desenvolvida neste trabalho envolve, primeiro, a divisão dos nós da RSSF em duas categorias, uma de maior e a outra de menor criticidade. Tomando como base a tabela da ISA apresentada no Capítulo 2.1, Criticidade de Eventos em Ambientes Industriais, podemos considerar que aplicações de segurança e controle são de maior criticidade e monitoramento, geralmente em maior número, as de menor criticidade.

Partindo da premissa que todas as RSSIs são conhecidas, é possível estimar a PER de cada link e então aplicar uma metodologia de criação de rotas. Estas devem ser determinadas de modo a não comprometer a confiabilidade em comunicações críticas e atender ao tempo máximo de latência imposto pela rede.

4.3.1 Aplicações de Maior Criticidade

Tratando-se de aplicações de segurança e controle, a confiabilidade da comunicação é fator crucial. A probabilidade de perda de pacotes deve ser a menor possível. Por isso, as rotas atribuídas para estes nós terão como único objetivo realizar sua comunicação com a base do modo mais confiável. Estas rotas podem ser ponto a ponto ou múltiplos saltos, dentre todas as opções, a que tiver menor estimativa da PER será a escolhida.

A metodologia desenvolvida para a criação deste tipo de rota começa com a identificação dos enlaces de melhor RSSI com a base, e conseqüentemente a menor PER. Caso a melhor RSSI seja com o nó da rota em questão, a comunicação é ponto a ponto e a rota está definida. Caso não seja, é preciso analisar a possibilidade de se conseguir uma PER menor com múltiplos saltos. Para isso, estima-se a PER das rotas com os nós de melhor RSSI com a base como nós repetidores. Dentre todas as possibilidades analisadas, a com menor RSSI é escolhida.

Tomando como exemplo uma RSSF com 5 nós e que o nó 4 seja de alta criticidade. As RSSIs desta rede estão apresentadas na Tabela 8, apesar de ser

usado apenas para exemplificar a metodologia, estes valores foram extraídos experimentalmente

Tabela 8 – RSSIs Entre Nós, Exemplo

-	Para	0	1	2	3	4	5
De	-						
0		-	-75,2	-61,4	-62,85	-61,8	-64,7
1		-70,75	-	-62,6	-75,6	-56,25	-70,9
2		-63,15	-69,35	-	-55,7	-63,7	-55,7
3		-62,7	-79,35	-54,35	-	-49,95	-62,9
4		-63,2	-63,05	-63,04	-52,05	-	-57,06
5		-65,55	-75,35	-54,85	-64,45	-56,7	-

Com esta tabela é possível classificar os nós que possuem melhores RSSIs com a base e classifica-los de acordo com seus valores. Deste modo chega-se a Tabela 9.

Tabela 9 – Classificação de RSSIs, Base

	1	2	3	4	5
De 0	5ª	1ª	3ª	2ª	4ª
Para 0	5ª	2ª	1ª	3ª	4ª

Nota-se que o nó com melhor RSSI de *downlink* com a base é o 2 e o de melhor *uplink* é o 3. Como o nó da análise em questão é o 4, existe a possibilidade de se conseguir uma PER menor com múltiplos saltos. A partir desta tabela, de classificação dos melhores links referente a base, forma-se o primeiro de dois conjuntos de possíveis rotas de menor PER. As rotas candidatas a serem escolhidas são as com os seguintes endereços:

- 0-4-0
- 0-3-4-3-0
- 0-2-4-2-0

Estas rotas são formadas inserindo os nós com links mais bem classificados que o nó em questão. Neste caso a base tem a melhor RSSI com 2 e os nós com melhor RSSI com a base são 2 e 3. O segundo conjunto de possíveis rotas de menor PER é formado partindo dos links mais bem classificados referentes ao nó 4. A Tabela 10 mostra a classificação destes links.

Tabela 10 – Classificação de RSSIs, Nó 4

	0	1	2	3	5
De 4	5ª	4ª	3ª	1ª	2ª
Para 4	4ª	2ª	5ª	1ª	3ª

Apesar de 4 estar bem classificada na tabela com a base como referência, a base não está bem classificada na tabela com 4 como referência. A RSSI de 4 para a base é a pior dentre todas as RSSIs que partem de 4. A RSSI da base para 4 é a segunda pior dentre as RSSIs em que 4 é o destino. O fato da RSSI de 2 para 4 ser pior que da base para 4 elimina a possibilidade de que as rotas que percorram este link sejam de menor PER. Portanto restam as seguintes rotas candidatas a menor PER:

- 0-1-4-1-0
- 0-5-4-5-0

Para descobrir qual destas é a rota mais confiável é necessário fazer a estimativa da PER pela RSSI.

4.3.2 Aplicações de Menor Criticidade

Em aplicações de monitoramento, o roteamento passa a atender ao propósito de diminuir a latência da rede em detrimento da confiabilidade da comunicação. A diminuição do número de saltos, ou envios de pacotes, é o método abordado por este trabalho, conforme apresentado no Capítulo 3.2, Determinação de Melhores Rotas.

O processo de seleção de melhores rotas passa primeiro no agrupamento dos nós em grupos de mesma frequência de aquisição de dados. Os nós com mesmos intervalos de aquisição poderão compartilhar da mesma rota. Nós com frequências de aquisição distintas devem também ter rotas distintas. A rede pode ser configurada com a estratégia de comunicação ponto a ponto ou múltiplos saltos. Neste segundo caso todos os nós da rota também serão nós destinos do pacote, não apenas retransmissões.

Para elencar as rotas elegíveis a terem menor latência considera-se a possibilidade de comunicar todos os nós com uma ou mais rotas. O número de rotas possíveis será todo valor múltiplo do número total de nós do grupo. Por exemplo, caso o grupo tenha 6 nós, é possível criar uma rota que abrange todos os 6, duas rotas com 3 nós, 3 rotas com 2 nós ou 6 rotas com 1 nó cada (ponto a ponto). Vale ressaltar que, se o total de nós for um número primo, só existirão as possibilidades de ter uma rota com todos os nós ou múltiplas rotas ponto a ponto. É preciso desenvolver todas as possibilidades para, posteriormente, aferir qual delas é a de menor latência.

O método para encontrar as sequência de nós que farão parte da rota é semelhante ao método para aplicações de alta criticidade. Começa com a classificação dos nós de melhor RSSI com a base. Os nós com melhor RSSI de envio e recepção com a base serão, respectivamente, o começo e o fim da rota. Iniciando com os endereços dos extremos, os nós intermediários serão escolhidos sequencialmente pelo critério de melhor RSSI até que todos os nós tenham sido abordados.

Para exemplificar como estas rotas de fato são criadas, este método pode ser demonstrado dando sequência ao mesmo cenário apresentado para aplicações de alta criticidade. Uma RSSF com cinco nós, sendo que quatro deles são para monitoramento. Neste caso existe a possibilidade de uma rede ponto a ponto, com um ou dois anéis, cada um referente a uma rota. Começando pela criação das duas elegíveis rotas da arquitetura em anel simples. A Tabela 8, apresentada no exemplo anterior, mostra que o nó 2 tem melhor RSSI de envio da base e o 3 de melhor recepção. Portanto estes serão os nós de início de criação da rota, os próximos nós que serão escolhidos serão aqueles de melhor RSSI de envio de 2 e de recepção de 3. A Tabela 11 classifica os nós de melhor RSSI com estes dois.

Tabela 11 – Classificação de RSSIs, Nó 2

Para	0	1	3	4	5
De 2	3ª	5ª	1ª	4ª	1ª

Os nós 3 e 5 possuem a mesma RSSI de 2, as duas possibilidades deveriam ser testadas, mas 3 já faz parte da rota e não pode ser escolhido, portanto apenas

5 é considerado. O nó 3 por sua vez só pode receber de 1. Forma-se, então, a primeira rota, 0-2-5-1-3-0.

A segunda rota elegível é formada com os mesmos princípios que a primeira, porém, ao invés de começar a escolha pelo sucessor de 2, começará pelo antecessor de 3. Para isso temos a Tabela 12 que apresenta as classificações dos nós com melhor RSSI para 3.

Tabela 12 – Classificação de RSSIs, Nó 3

De	0	1	2	4	5
Para 3	3ª	5ª	2ª	1ª	4ª

Os três primeiros nós da tabela de prioritários de envio para 3 são os 4, 2 e 0, todos impossibilitados de serem escolhidos. O 4 por ser de outra categoria de criticidade, 2 por já fazer parte da rota e a base por ainda existir nós para compor a rota. Então o nó 5, que é o quarto colocado, assume a posição de antecessor de 3 na rota. O sucessor de 2 só pode ser 1, que é o único que falta. Forma-se assim a segunda rota elegível, 0-2-1-5-3-0.

Existe também a possibilidade de se formar um par de rotas com dois nós em cada. A formação destas rotas tem como base as rotas que já foram criadas. Cada par de rotas elegíveis surge da divisão das rotas já criadas. Neste caso são criados os dois pares de rotas, 0-2-5-0 com 0-1-3-0 e 0-2-1-0 com 0-5-3-0. Estas rotas devem ser tratadas, necessariamente, como pares pois trata-se de rotas complementares. A latência da rede dedicada a este grupo será a soma das latências do par.

A outra arquitetura possível é a ponto a ponto, neste caso não é necessário que se utilize de nenhuma estratégia. Criam-se as rotas conectando um nó por rota com a base, formando um grupo de rotas que devem ser tratados como conjunto. Neste caso as rotas deste conjunto são 0-1-0, 0-2-0, 0-3-0 e 0-5-0. Para deliberar sobre a escolha dentre as possíveis alternativas é preciso aferir a latência resultante para cada uma.

5 Material e Métodos

Neste trabalho foram propostos dois experimentos, cada um com um objetivo. O primeiro é mensurar o valor da PER para determinadas RSSIs. O segundo é implementa o algoritmo para coletar as RSSIs em um cenário real.

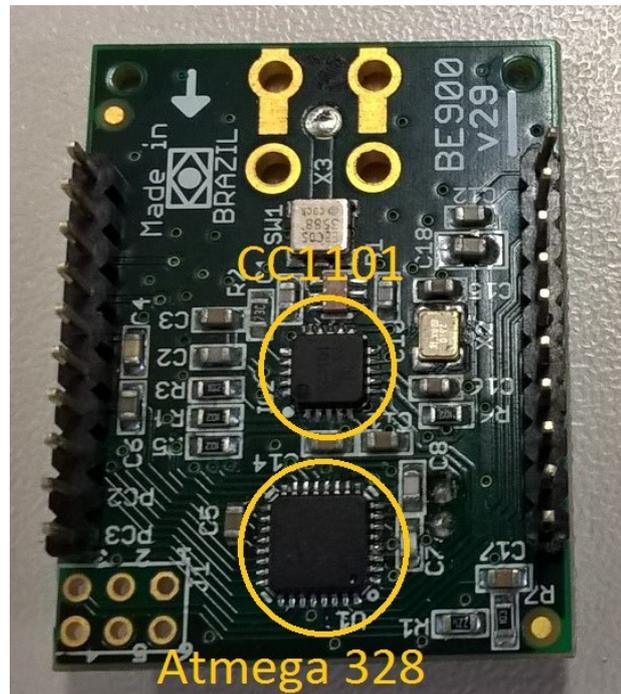
O *hardware* utilizado é o mesmo para ambos os experimentos, este é composto pelo nó sensor BE900 [BE900], alimentado pela placa DK102 e o modulo UartsBee para comunicação com o software gerente.

Os experimentos não compartilham do mesmo *software* nem do mesmo *firmware* dos nós sensores. Ambos os *softwares* foram desenvolvidos em Phyton e *firmwares* pela plataforma Radiumino (BRANQUINHO, 2014), cada um customizado a seu modo.

5.1 Hardware

O modulo de comunicação sem fio utilizado nos experimentos foi o BE900, utilizando tanto como nó sensor como a base. Composto por um processador AVR Atmega328 e um transceptor TI CC1101 RF com filtro passa-faixa ajustado para operar na banda ISM de 915 MHz (902-907,5MHz e 915-928MHz). Este módulo é compatível com a plataforma Radiumino podendo ser programado utilizando pelo IDE Arduino, usufruindo da maioria dos softwares e bibliotecas a ele destinados. Suas dimensões são 24,4mm por 32mm e 10,5mm de altura e possui um total de 20 pinos. A Figura 20 ilustra o BE900.

Figura 20 – BE900

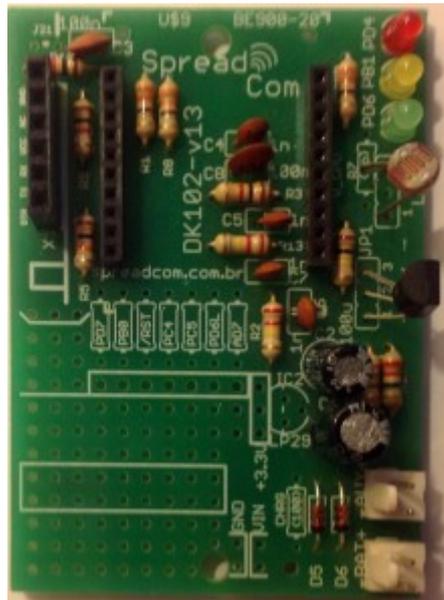


As características de transmissão que foram atribuídas ao BE900 foram:

- Taxa: 4,8Kbps
- Potência de transmissão: -30dBm para aferição da PER e 10dBm na implementação do protocolo de coleta de RSSIs
- Modulação: 2FSK
- Canal: 0
- Banda do canal: 541,66KHz

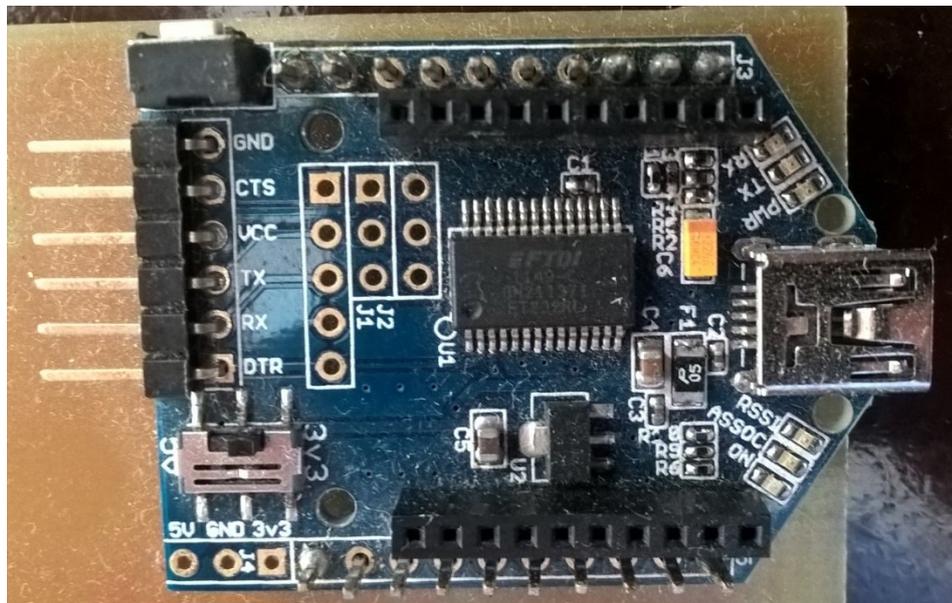
Para alimentar os módulos utilizados como nós sensores foram utilizadas as placas do kit de desenvolvimento DK102. Esta placa possui um regulador de tensão que permite que a fonte de alimentação seja de 5V a 14V, possibilitando a utilização de diferentes fontes. Seus LEDs são úteis para indicar a transmissão, recepção e a alimentação do módulo. Apesar de contar com sensores de temperatura e luminosidade, estes não foram utilizados. A Figura 21 ilustra esta placa.

Figura 21 – DK102



O DK102 não possui conversor para comunicação Serial USB e por isso não pôde ser utilizado na base. Neste caso foi utilizado o módulo UartSBee, este possui o soquete compatível com o BE900 e se comunica via USB por um conector mini USB fêmea. A figura 22 representa este módulo.

Figura 22 - UartsBee



Um cabo USB-mini USB conecta o UartsBee ao notebook responsável por rodar o arquivo em Python que faz a gerência da rede.

Para o experimentos de aferição da PER também foram utilizado atenuadores fixos de alta precisão fabricado pela *Mini-Circuits*, apresentado na Figura 23. Estes elementos possuem conectores SMA e diferentes atenuações, 3dB, 6dB, 10dB, 20dB e 30dB.

Figura 23 – Atenuadores Fixos



5.2 Firmware

Os firmwares foram desenvolvidos utilizando a plataforma Rádium. Esta é uma plataforma para prototipagem de software aberto, baseada em Arduino e adaptada para RSSF. Composta por um conjunto de *Hardware* e *Software*, implementados sob um microcontrolador MegaAVR da Atmel e um transceptor CC1101 da *Texas Instrument*, a plataforma permite ainda incorporar outros módulos de *hardware* de acordo com a aplicação, como sensores e atuadores (BRANQUINHO, 2014).

A estrutura do protocolo de comunicação do Rádium é formada por uma pilha com cinco camadas: Física, MAC, Rede, Transporte e Aplicação. O pacote de comunicação é composto por 52 bytes, sendo 4 para cada uma das três primeiras camadas e 36 bytes para a aplicação. O usuário tem acesso livre a estas camadas,

podendo alterar seus códigos e criar uma rede com as características desejadas. Segundo PRESSMAN (2006), as funcionalidades implementadas em uma camada devem ser limitadas pelas responsabilidades a ela atribuídas, facilitando reuso e manutenção das funções desenvolvidas. Contudo, a aplicação do conceito de *cross-layer* redefine as responsabilidades de cada camada, uma vez que permite a atuação de camadas inferiores na tomada de decisão de camadas superiores. De acordo com SRIVASTAVA (2005), soluções *cross-layer* tendem a ser exclusivas, desenvolvidas para resolver problemas específicos.

5.2.1 Firmware para Aferição da PER

Neste experimento foram utilizados apenas dois módulos Be900, cada um com um firmware dedicado a ele. O firmware desenvolvido para o transmissor realiza o envio e a contagem de pacotes enviados, esta informação vai dentro do próprio pacote sendo que 6 dos 52 bytes foram destinados para isso. O firmware desenvolvido para o receptor realiza a recepção, medição da RSSI, contagem de pacotes recebidos e retransmissão dos pacotes via USB, outros 6 bytes do pacote foram destinados a contagem.

A taxa de transmissão utilizada neste experimento foi de 4,8Kbps. O tamanho total do pacote é de 64 bytes, considerando também os bytes de sincronismo, ou 512bits. Esta configuração resultam no tempo de transmissão de 0,107 segundos, para respeitar este tempo e evitar possíveis conflitos no processamento, o transmissor foi programado para transmitir pacotes a cada 0,2 segundos. O mapa do pacote estabelecido para este experimento esta apresentado na Tabela 13.

Tabela 13 – Mapa do Pacote, Firmware para Aferição da PER

Física				MAC				
RSSI	LQI	-	-	-	-	-	-	
1	2	3	4	5	6	7	8	
Rede				Transporte				
-	-	-	-	-	-	-	-	
9	10	11	12	13	14	15	16	
Aplicação								
-	-	-	-	-	-	-	-	-
17	18	19	20	21	22	23	24	25
Aplicação								
-	-	-	-	-	-	-	-	-
26	27	28	29	30	31	32	33	34
Aplicação								
CONT ENV 1	CONT ENV 2	CONT ENV 3	CONT ENV 4	CONT ENV 5	CONT ENV 6	CONT REC 1	CONT REC 2	CONT REC 3
35	36	37	38	39	40	41	42	43
Aplicação								
CONT REC 4	CONT REC 5	CONT REC 6	-	-	-	-	-	-
44	45	46	47	48	49	50	51	52

O pacote possui a mesma divisão em camadas pré-definidas pela plataforma RADIOLINK. Os valores de RSSI e LQI da recepção do pacote são alocados nos dois primeiros bytes. Os contadores de envio de pacote foram alocados do byte 35 ao 40 e os contadores de recepção do byte 41 ao 46. Neste experimento não foi necessário endereçar o pacote.

5.2.2 Firmware do Protocolo de Coleta de RSSIs

O protocolo implementado foi no modo de cálculo de destino realizado pela nó sensor de envio e sem as rotas alternativas. A adaptação do firmware padrão para a implementação do protocolo foi feita nas camadas de rede e aplicação. Em rede foi inserido o código para calcular o nó sensor de destino e em aplicação foi feito a alocação das RSSIs de acordo com o salto da rota. Esta alocação exigiu 3 bytes para cada salto, um para o endereço do nó sensor, um para a RSSI e um para a LQI (são as duas métricas fornecidas pela camada física, a LQI não foi utilizada neste trabalho mas foi implementada para testes futuros). Por isso, este

firmware permite que o protocolo seja implementado em uma rede com até 12 nós sensores. O mapa do pacote utilizado neste experimento esta apresentado na Tabela 14.

Tabela 14 – Mapa do Pacote, Firmware do Protocolo de Coleta

Física				MAC				
RSSI 0	LQI 0	-	-	-	-	-	-	
1	2	3	4	5	6	7	8	
Rede				Transporte				
ID DESTINO	SALTOS	-	TOTAL DE NÓS	-	-	-	-	
9	10	11	12	13	14	15	16	
Aplicação								
RSSI 1	LQI 1	ID 1	RSSI 2	LQI 2	ID 2	RSSI 3	LQI 3	ID 3
17	18	19	20	21	22	23	24	25
Aplicação								
RSSI 4	LQI 4	ID 4	RSSI 5	LQI 5	ID 5	RSSI 6	LQI 6	ID 6
26	27	28	29	30	31	32	33	34
Aplicação								
RSSI 7	LQI 7	ID 7	RSSI 8	LQI 8	ID 8	RSSI 9	LQI 9	ID 9
35	36	37	38	39	40	41	42	43
Aplicação								
RSSI 10	LQI 10	ID 10	RSSI 11	LQI 11	ID 11	RSSI 12	LQI 12	ID 12
44	45	46	47	48	49	50	51	52

Os dois primeiros bytes são reservados para os valores de RSSI e LQI do pacote recebido pela base. Na camada de rede encontram-se os bytes referentes as informações necessárias para a recepção e retransmissão do pacote. Todos a camada de aplicação é destinada aos valores de RSSI, LQI e endereço dos nós que compõem a rota.

O *cross-layer* foi necessário nas alocações da camada de aplicação por dois motivos. O primeiro é que o número do byte depende do número de saltos, camada de rede, por exemplo, os valores de RSSI, LQI e endereço referentes ao quinto salto são alocados nos bytes 29, 30 e 31, respectivamente. O segundo motivo é que os próprios valores que são alocados pertencem a outras camadas, RSSI e LQI na camada de rede e endereço na camada de rede.

5.3 Softwares

Os softwares para a execução dos experimentos foram desenvolvidos em Python e possuem funções bem distintas que serão tratadas nos próximos tópicos.

5.3.1 Software para Aferição da PER

Para o experimento de aferição da PER, o software desenvolvido possui sete funções. A primeira delas é abrir uma porta de comunicação com a base, conectada via USB, o número desta porta deve ser informada pelo usuário. A segunda função é ler os pacotes recebidos pela porta que foi aberta. A terceira é converter o byte referente a RSSI em seu valor físico, em dBm, conforme determinado pelo *datasheet* do transceptor CC1101 [REF]. Esta conversão pode ser representada pela função apresentada abaixo.

$$f(RSSI) = \begin{cases} RSSI = \left(\frac{RSSIC - 256}{2} \right) - 74, & RSSI > 128 \\ RSSI = \left(\frac{RSSIC}{2} \right) - 74, & RSSI \leq 128 \end{cases}$$

Sendo RSSIC o valor informado pelo transceptor e a RSSI é seu valor em dBm. A quarta função deste software é transformar os dois contadores, de 6 bytes cada, em valor decimal. O cálculo desta transformação está representado pela equação abaixo.

$$Cont = B1 + (B2 * 256) + (B3 * 256^2) + (B4 * 256^3) + (B5 * 256^4) + (B6 * 256^5)$$

Sendo Cont, o valor do contador em decimal e cada um dos Bs, um byte contador do pacote.

A quinta função é calcular a quantidade de pacotes perdidos. Este valor é encontrado pela subtração dos pacotes enviados pelos pacotes recebidos. A sexta é mostrar na tela em tempo real os valores de RSSI e o número de pacotes enviados, recebidos e perdidos. A sétima e última função deste software é criar e

armazenar em um arquivo texto os valores que foram impressos na tela e o horário em que o pacote foi recebido.

5.3.2 Software Gerente do Protocolo de Coleta de RSSIs

Neste experimento, o software desenvolvido trata-se do software gerente, responsável pela gestão da rede. A ele foram atribuídas seis funções. Para executar suas funções é preciso alimentá-lo com as informações do total de nós sensores da rede, o número de ciclos de envio de pacotes e o número porta que faz a comunicação serial com a base.

Assim como no experimento anterior, a primeira de suas funções é abrir uma porta de comunicação com a base. A segunda função é a de disparo de pacotes. Esse disparo é realizado incrementando o byte de endereço de destino, do 1 ao último nó da rede, informação dada pelo usuário. A terceira função deste software é ler os pacotes que forem recebidos pela porta de comunicação aberta com a base. A quarta é converter todos os bytes referentes a valores de RSSI em valores físicos, como no experimento anterior. Neste caso são 14 bytes destinados à RSSI que devem ser convertidos, lembrando que, a cada salto dado, uma nova RSSI é alocada no pacote. A quinta função é de mostrar na tela, em tempo real, se o pacote foi recebido ou foi perdido. A sexta e última função é armazenar em um arquivo texto os valores de RSSI e endereços de todos os nós sensores que transmitiram o pacote. O ciclo de envio e leitura do pacote será executando quantas vezes for requisitado.

5.4 Bancada de Testes para Aferição da PER pela RSSI

O ambiente experimental desenvolvido para os testes é composto por dois módulos transceptores, uma câmara blindada, atenuadores e um computador, apresentados na Figura 24. Um dos módulos fica dentro da câmara blindada, este é responsável pela transmissão sucessiva de pacotes. O segundo módulo fica do lado de fora e conectado ao computador via USB. Os dois módulos se comunicam via cabo coaxial.

Figura 24 – Bancada de Testes para Aferição da PER



Em destaque em amarelo na figura está a base. Os pacotes disparados pelo nó dentro da caixa blindada e recebidos pela base é encaminhado ao computador que executa o software de coleta. Como as contagens são realizadas nos próprios módulos, os pacotes eventualmente perdidos entre a base e o computador não serão contabilizados, isolando este problema. Foi considerado que um universo amostral de 250 mil transmissões seja suficiente para se estimar a PER com razoável precisão, aproximadamente 14 horas de coleta para cada teste.

5.5 Implementação do Protocolo de Coleta

A implementação do protocolo de coleta de RSSI foi desenvolvida em uma rede com dez nós sensores. Três deles alimentados via adaptador de tomada e sete por baterias. Em todos os nós foram utilizadas antenas omnidirecionais. A Figura 25 apresenta o material utilizado.

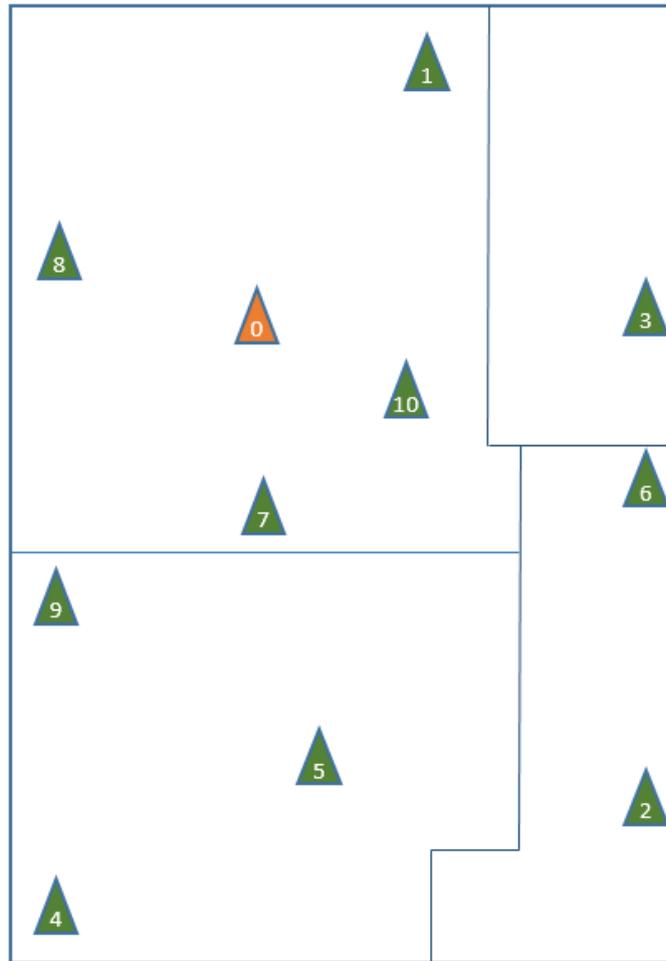
Figura 25 – RSSF da Implementação do Protocolo de Coleta



5.5.1 Cenário de Teste

A RSSF foi instalada no laboratório de Meios de Transmissão da PUC Campinas. O local é dividido em três repartições, uma sala de computadores, uma de bancadas e uma de suporte, separadas por divisórias e portas de PVC e vidro. A distribuição dos nós foi feita com a intenção espalha-los na região para obter valores de RSSI o mais variados quanto possível. A Figura 26 trata-se do mapeamento dos nós sensores no laboratório em questão.

Figura 26 – Mapa da Distribuição de Nós



Os nós foram colocados, de forma aleatória, em diferentes alturas, a utilização de baterias permitiu que os locais fossem escolhidos independentemente da existência de pontos de força. A Figura 27 trata-se de um conjunto de fotos que mostram as posições de cada nó no experimento.

Figura 27 – Cenário Implementado



6 Análise de Resultados

6.1 Aferição da PER

Os experimentos foram realizados com cinco atenuações diferentes, resultando em potências médias de recepção iguais a -82,8dBm, -87dBm, -95,3dBm, -99,2dBm e -100,7dBm. Os experimentos obtiveram comportamentos semelhantes com relação a variação de RSSI e ocorrências de percas de pacote.

6.1.1 Validação dos Resultados

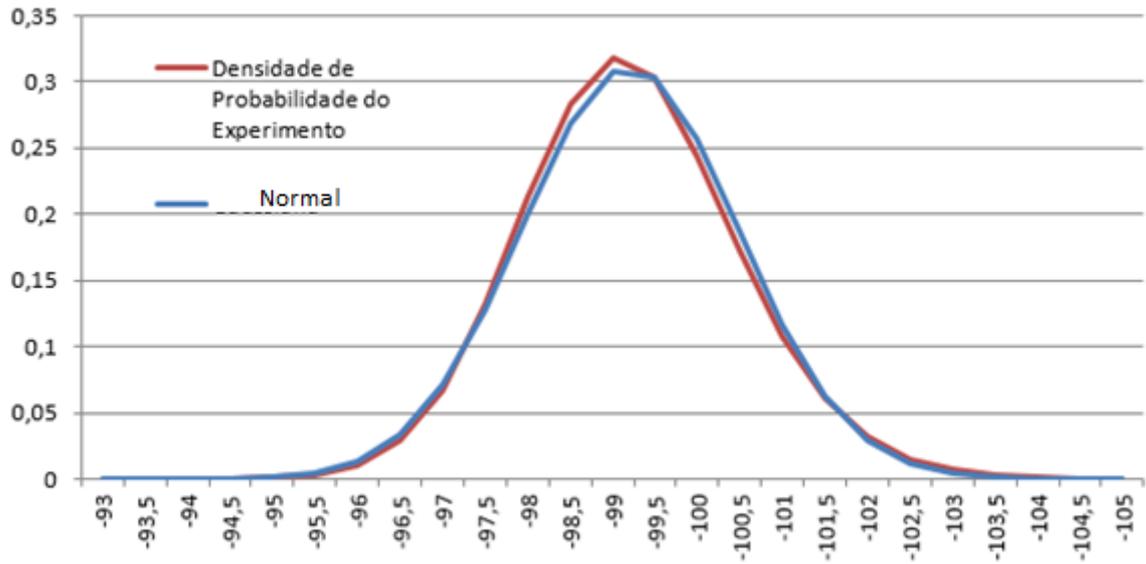
Tomemos o experimento com a potência de recepção igual a -99,2dBm como exemplo para ilustrar as análises de validação do experimento. Neste experimento foram transmitidos um total de 297795 pacotes. É esperado que a RSSI varie ao longo do tempo, esta variação é causada por fontes de ruídos, como o AWGN, que não foram possíveis de se isolar. Esta variação deve seguir uma distribuição padrão, sem eventos isolados que podem corromper a confiabilidade do experimento. Devido as fontes de ruído, a distribuição que tende a representar melhor a variação da RSSI em um ambiente isolado é a Distribuição Gaussiana, ou Normal.

A função Densidade de Probabilidade da Gaussiana é representada pela seguinte equação.

$$f(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)}$$

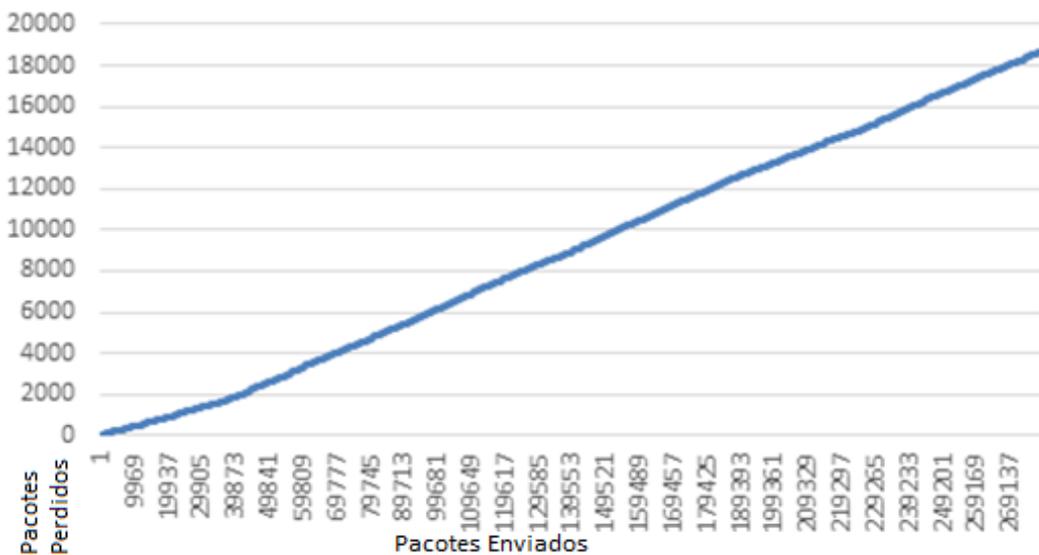
Sendo média μ , desvio padrão σ e variável aleatória x . No exemplo do experimento em questão, o desvio padrão da RSSI é de 1,018 e sua média é de -99,2dBm. A Figura 28 apresenta as curvas Normal e da densidade de probabilidade do experimento.

Figura 28 – RSSI X Normal e Resultados Experimentais



Nota-se que as duas curvas são muito próximas, indicando que o experimento não sofreu nenhuma anomalia durante a coleta. Outro gráfico relevante na validação do experimento é da curva do número de pacotes perdidos ao longo do tempo. Este gráfico está representado pela Figura 29.

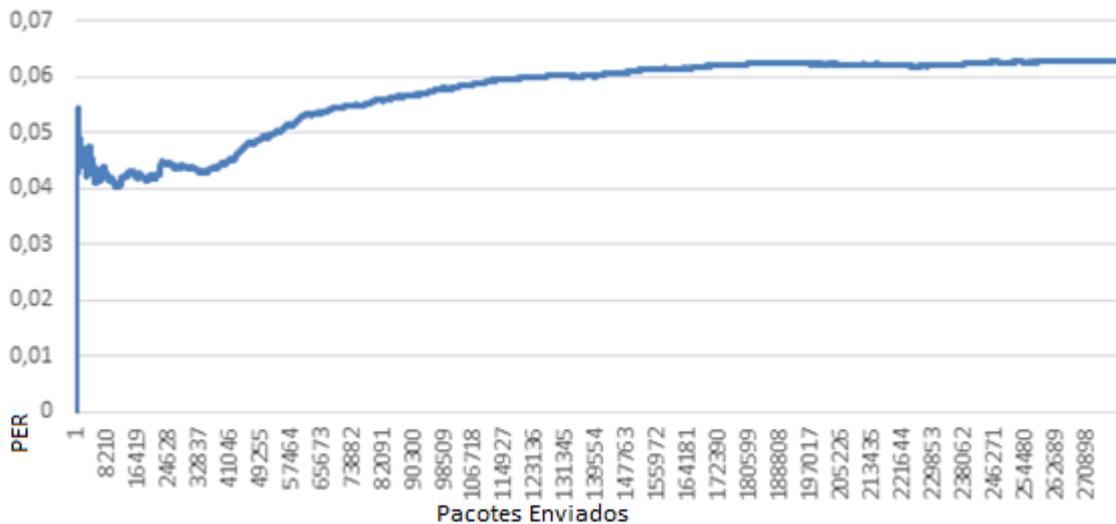
Figura 29 – Pacotes Enviados X Pacotes Perdidos



A linearidade da variação do número de pacotes perdidos ao longo do tempo indica uma certa previsibilidade de ocorrer a perda. Para averiguar se a quantidade de amostras foi suficiente para que se obtenha um valor de PER confiável, é preciso

analisar o comportamento de sua variação durante a coleta. O gráfico da variação da PER ao longo do tempo está representado pela Figura 30.

Figura 30 – Pacotes Enviados X PER



Observando este gráfico nota-se que o valor da PER atingiu uma razoável estabilidade ao final da coleta, indicando que a quantidade de amostras é suficiente para este propósito.

6.1.2 Estimativa da PER em Função da RSSI

Os resultados obtidos com este experimento demonstram uma evidente relação entre a PER e a RSSI. A relação de valores encontrado está apresentado pela Tabela 15.

Tabela 15 – Resultado do Experimento para a Estimativa da PER

PER	RSSI (dBm)
0,00038	-82,8
0,00043	-87
0,00092	-95,3
0,06273	-99,2
0,26017	-100,7

Nota-se que quanto menor a RSSI, maior a PER. Utilizando-se da curva Eb/No pela PER, encontra-se o valor da RSSI equivalente ao Eb/No para uma dada

PER. A Tabela 16 apresenta os valores de Eb/No e RSSI que possuem a mesma PER.

Tabela 16 – Comparação entre RSSI e Eb/No

RSSI (dBm)	Eb/No
-82,8	14,3
-87	14,2
-95,3	14
-99,2	12,2
-100,7	11,3

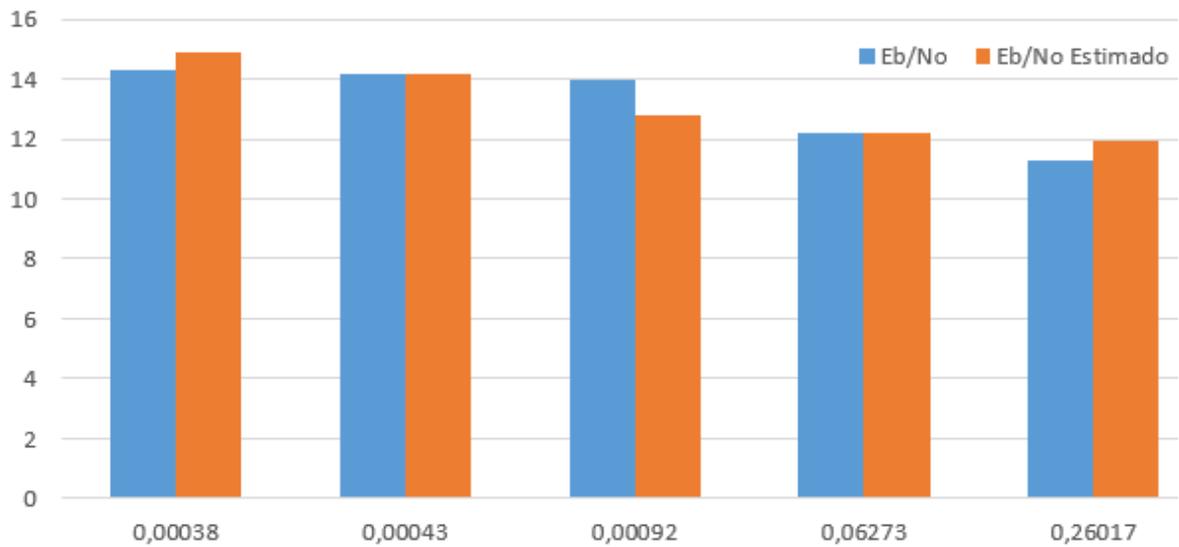
Analisando os valores extremos de ambos parâmetros, observa-se que a diferença entre o valor máximo e mínimo da RSSI é de 17,9 dB e de Eb/No é 3. A diferença da RSSI é de aproximadamente 6 vezes maior que Eb/No. Então, dividindo o valor da RSSI por 6 e calculando a diferença entre o valor encontrado e EB/No tem-se a Tabela 17.

Tabela 17 – Diferença entre Eb/No e RSSI Ajustada

Eb/No	RSSI / 6	Diferença
14,3	-13,8	28,1
14,2	-14,5	28,7
14	-15,9	29,9
12,2	-16,5	28,7
11,3	-16,8	28,1

Calculando a média da diferença entre os valores encontra-se 28,7. Somando este valor a RSSI/6, chega-se a uma aproximação entre a RSSI e Eb/No. Os valores encontrados estão representados pelo gráfico da Figura 32.

Figura 31 – PER X Eb/No e Eb/No Estimado

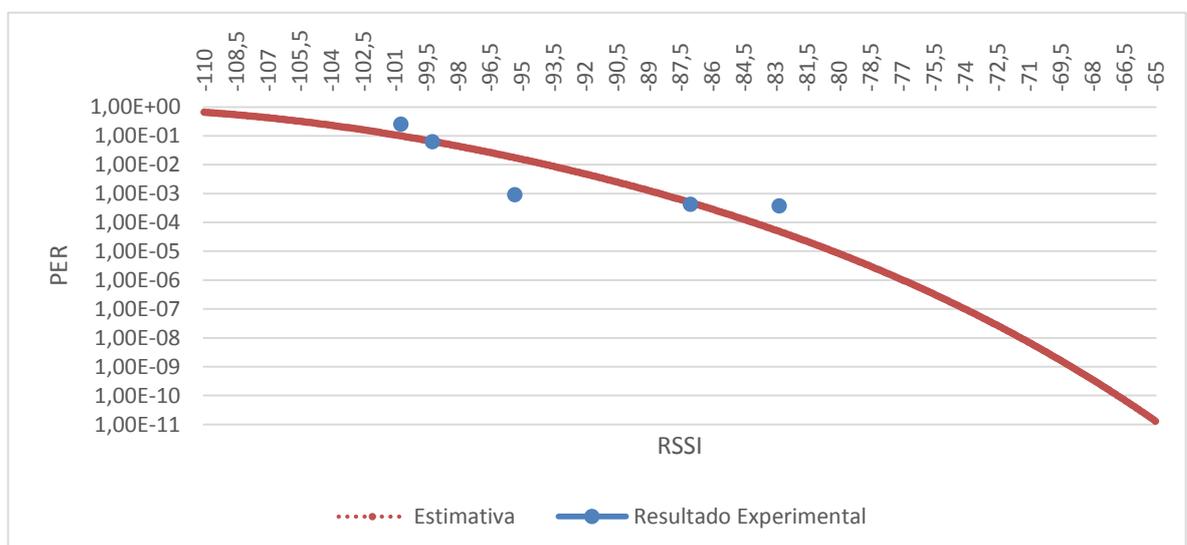


Nota-se uma razoável aproximação entre a estimativa de Eb/No a partir da RSSI e o valor de Eb/No correspondente a PER indicada. Utilizando das etapas que foram apresentadas, encontra-se a seguinte equação que relaciona o valor de Eb/No com a RSSI.

$$\frac{Eb}{No} = \frac{RSSI}{6} + 28,7$$

Fazendo a conversão da RSSI e utilizando a equação que relaciona Eb/No com a PER, demonstrada anteriormente, chega-se a curva da estimativa da PER pela RSSI, apresentada na Figura 32.

Figura 32 – RSSI X PER Estimada e PER Medida



6.2 Coleta de RSSIs

Com o experimento realizado foi possível coletar todas as intensidades de sinais entre todos os nós da rede em um ambiente real. Estas RSSIs são suficientes para fazer uma análise bem fundamentada sobre a configuração de rotas com as metodologias utilizadas neste trabalho. Ao todo foram coletadas as intensidades de sinal de 90 links diferentes. A análise foi feita em cima da média móvel realizada com as últimas 10 amostras de cada link. Devido a variação de sinal, a utilização da média móvel das últimas amostras se torna conveniente por ser menos susceptível a falhas. O valor da média da intensidade de sinal das últimas 10 amostras esta apresentada na Tabela 18.

Tabela 18 – RSSIs Coletadas pelo Protocolo de Coleta

		RSSI										
-	Para	0	1	2	3	4	5	6	7	8	9	10
De	-											
0		-	-52,85	-67,15	-68,35	-82,55	-63,1	-58,3	-53,35	-54,05	-53,05	-40,7
1		-50,25	-	-72	-67,65	-78,65	-75,45	-70,8	-60,75	-41,75	-52,65	-55,6
2		-66,25	-73,7	-	-62,3	-61	-50,35	-58	-59,45	-75,7	-54,7	-53,75
3		-65,85	-67,95	-59,8	-	-75,2	-61,4	-62,85	-61,8	-64,7	-69,1	-63,1
4		-73,1	-74,25	-53,9	-70,75	-	-62,6	-75,6	-56,25	-70,9	-47,45	-60,25
5		-62	-77,35	-49,55	-63,15	-69,35	-	-55,7	-63,7	-55,7	-55,6	-59
6		-55,5	-71,8	-55,1	-62,7	-79,35	-54,35	-	-49,95	-62,9	-61,55	-61,25
7		-53,75	-61,95	-58,45	-63,2	-63,05	-63,04	-52,05	-	-57,06	-56,05	-45,35
8		-52,55	-42,55	-74,3	-65,55	-75,35	-54,85	-64,45	-56,7	-	-54,6	-56,85
9		-50,1	-52	-51,2	-68,15	-51,1	-53,2	-61,7	-52,4	-53,2	-	-57
10		-36,05	-53,45	-48,45	-60,05	-61,85	-54,65	-56,9	-40,5	-53,75	-54,65	-

Com estes valores é possível estabelecer as rotas elegíveis utilizando os mesmos princípios apresentados neste trabalho. Aplicando o método de estimativa da PER, desenvolvida neste trabalho, em todas as RSSIs coletadas, encontra-se a Tabela 19.

Tabela 19 – PERs Estimadas pelas RSSIs Coletadas

ESTIMATIVA DA PER												
-	Para	0	1	2	3	4	5	6	7	8	9	10
De	-											
0	-	0	1,5E-10	5,3E-10	4,3E-05	1,3E-12	0	0	0	0	0	0
1	0	-	1,8E-08	2,5E-10	3,5E-06	3,2E-07	5,9E-09	5,7E-14	0	0	0	0
2	5,5E-11	7,8E-08	-	4,5E-13	5,7E-14	0	0	0	3,9E-07	0	0	0
3	3,5E-11	3,5E-10	0	-	2,7E-07	1,7E-13	9,7E-13	2,3E-13	9,3E-12	1,1E-09	1,3E-12	0
4	4,7E-08	1,2E-07	0	5,7E-09	-	6,8E-13	3,6E-07	0	6,5E-09	0	5,7E-14	0
5	3,4E-13	1,4E-06	0	1,4E-12	1,5E-09	-	0	2,8E-12	0	0	0	0
6	0	1,5E-08	0	8E-13	5,6E-06	0	-	0	1E-12	1,7E-13	1,1E-13	0
7	0	2,8E-13	0	1,5E-12	1,3E-12	1,2E-12	0	-	0	0	0	0
8	0	0	1,3E-07	2,5E-11	3E-07	0	6,9E-12	0	-	0	0	0
9	0	0	0	4,3E-10	0	0	2,3E-13	0	0	-	0	0
10	0	0	0	0	2,8E-13	0	0	0	0	0	-	0

A precisão do software utilizado para executar os cálculos vai até a vigésima casa decimal, portanto, valores menores que 10^{-20} são arredondados para 0. Este arredondamento, na prática, não é um problema. Por se tratar de uma gerência de RSSF, valores de PER tão baixos podem, de fato, ser considerados zero.

O primeiro passo para a criação de qualquer rota é definir os nós de maior e os de menor criticidade. Para exemplificar a metodologia foi considerado que dois dos nós sensores da rede são de alta criticidade, estes são os nós 4 e 7. A formação de qualquer rota depende da classificação dos nós com relação a suas RSSIs com a base. Esta classificação está representada pela Tabela 20.

Tabela 20 – Classificação de RSSIs, Base

	1	2	3	4	5	6	7	8	9	10
De 0	2	8	9	10	7	6	4	5	3	1
Para 0	3	9	8	10	7	6	5	4	2	1

6.3 Rotas criadas para aplicações de maior criticidade

Começando pela criação de rotas dos nós de maior criticidade. As primeiras rotas elegíveis serão de comunicação direta entre o nó e a base. Portanto, uma rota para o nó 4 e uma para 7, são elas:

- 0-4-0

- 0-7-0

O nó sensor 10 é aquele que possui melhores RSSIs com a base, portanto as próximas rotas elegíveis o terão como retransmissor. Cria-se então mais uma rota para cada nó, são elas:

- 0-10-4-10-0
- 0-10-7-10-0

As próximas rotas a serem criadas são aquelas que utilizam a classificação de melhores RSSIs do próprio nó. A tabela do nó 4 está apresentada pela Tabela 21.

Tabela 21 – Classificação de RSSIs, Nó 4

	0	1	2	3	5	6	7	8	9	10
De 4	8	9	2	6	5	10	3	7	1	4
Para 4	10	8	2	6	5	9	4	7	1	3

O nó sensor 9 possui as melhores RSSIs com relação ao nó 4, portanto a próxima rota terá este nó como retransmissor, esta é:

- 0-9-4-9-0

A Tabela 22 a tabela de classificação de melhores RSSIs com relação ao nó 7.

Tabela 22 – Classificação de RSSIs, Nó 7

	0	1	2	3	4	5	6	8	9	10
De 7	3	7	6	10	9	8	2	5	4	1
Para 7	4	8	7	9	5	10	2	6	3	1

O nó com melhores RSSIs com relação ao 7 é o 10, portanto este será retransmissor na próxima rota, esta é:

- 0-10-7-10-0

Esta rota já foi revelada anteriormente. Conclui-se, então, a criação das rotas elegíveis para os dois nós de maior criticidade, resultando em um total de cinco rotas, três para o nó 4 e duas para o nó 7. A escolha das melhores rotas depende

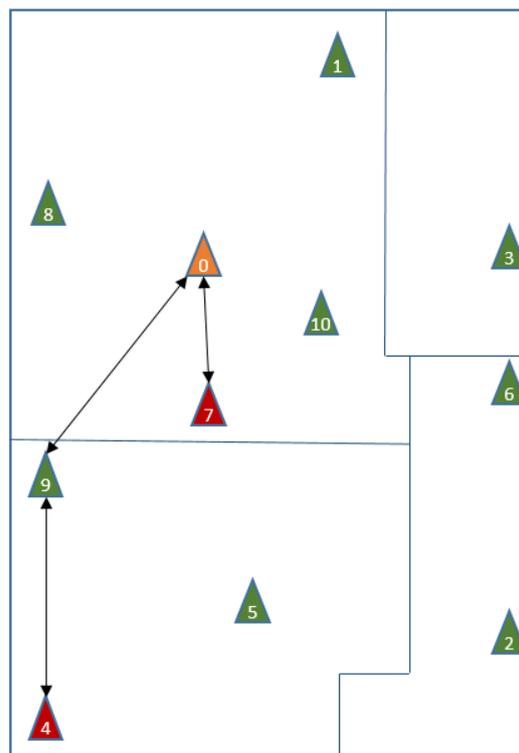
da estimativa da PER de cada uma, a que for menor é eleita. Realizando os cálculos de estimativa de PER, demonstrado no Capítulo XXX, temos as estimativas apresentadas na Tabela 23.

Tabela 23 – Rotas Elegíveis, Alta Criticidade

DESTINO	ROTA	PER
4	0-4-0	4,30E-05
	0-10-4-10-0	3,41E-13
	0-9-4-9-0	0
7	0-7-0	0
	0-10-7-10-0	0

As estimativas de PER das duas rotas referentes ao nó 7 são de aproximadamente 0, mas uma tem mais saltos que a outra. Utilizando o número de saltos como critério de desempate, a melhor rota para este é a 0-7-0. As estimativas referentes as rotas para o nó 4 são todas diferentes, uma destas possui uma PER de aproximadamente 0, a rota 0-9-4-9-0. Esta, portanto é a rota mais confiável seguida da 0-10-4-10-0 e por último a rota direta, 0-4-0. A Figura 33 ilustra as melhores rotas no ambiente de teste.

Figura 33 – Melhores Rotas



6.4 Rotas criadas para aplicações de menor criticidade

A metodologia para criação de melhores rotas em aplicações de menor credibilidade, abordada por este trabalho, define que a melhor possibilidade é aquela de menor latência. No cenário analisado, são oito nós sensores destinados a aplicações de menor criticidade. Por isso existem quatro possibilidades, ponto a ponto, anel, dois e quatro anéis. A Tabela 24 apresenta a aplicação da conversão da RSSI em PER e o cálculo da latência utilizando os critérios estabelecidos neste trabalho.

Tabela 24 – Latência Resultante, Ponto a Ponto

Ponto a Ponto					
ROTAS			PER	Latência	Latência Total
0	1	0	0	2,000000000000000	16,0000000015432
0	2	0	2,04579E-10	2,00000000040916	
0	3	0	5,65365E-10	2,00000000113073	
0	5	0	1,64846E-12	2,00000000000330	
0	6	0	0	2,000000000000000	
0	8	0	0	2,000000000000000	
0	9	0	0	2,000000000000000	
0	10	0	0	2,000000000000000	

O próximo conjunto de rotas abordado resulta em uma topologia com quatro anéis. Neste caso cada uma das rotas engloba dois nós sensores e se utiliza de três saltos para ser executada. A Tabela 25 apresenta as rotas formadas com suas respectivas PERs e latência aplicando o método para formação de rotas.

Tabela 25 – Latência Resultante. 4 Anéis

Quatro Anéis						
ROTAS			PER	Latência	Latência Total	
0	10	2	0	5,54223E-11	3,00000000016627	12,0000000010583
0	1	3	0	2,90129E-10	3,00000000087039	
0	9	5	0	3,41061E-13	3,00000000000102	
0	8	6	0	6,87805E-12	3,00000000002063	

Na topologia em dois anéis, cada rota é composta por quatro nós sensores e cinco saltos. A Tabela 26 apresenta os resultados desta topologia.

Tabela 26 – Latência Resultante, 2 Anéis

Dois Anéis									
ROTAS						PER	Latência	Latência Total	
0	10	2	5	8	0	0	5,000000000000000	10,00000000127980	
0	1	3	6	9	0	2,55966E-10	5,00000000127983		

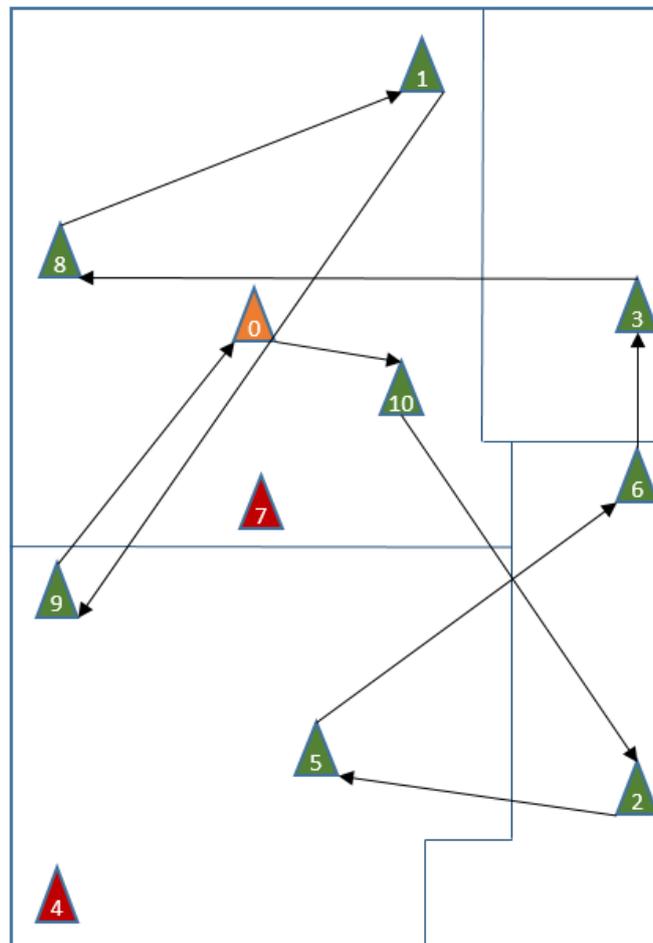
A última possibilidade é a topologia em anel simples. Os resultados desta topologia estão apresentados na Tabela 27.

Tabela 27 – Latência Resultante, Anel

Anel											
ROTA										PER	Latência
0	10	2	5	6	3	8	1	9	0	1,00613E-11	9,00000000009055

Das opções analisadas, a de menor latência total é a topologia em anel. A Figura 34 ilustra o percurso da rota no cenário de análise.

Figura 34 – Melhor Rota, Baixa Criticidade



7 Conclusão

Este trabalho desenvolveu três propostas distintas que interagem para um objetivo comum, a determinação de melhores rotas.

A primeira proposta é uma estimativa da PER pela RSSI. Esta estimativa utilizou como base a curva da BER de Eb/No para a modulação FSK, amplamente difundida na literatura. Os experimentos realizados tiveram como objetivo aferir a PER em diferentes intensidades de sinal. Utilizando a PER como valor comum, foi proposto uma formula que relaciona a RSSI com Eb/No.

A segunda proposta é um protocolo para coleta de RSSIs. Este é um protocolo de baixa complexidade computacional que permite coletar todas as RSSIs da RSSF. Sua operação se baseia em criação de rotas a partir de operações matemáticas básicas realizadas pelos nós. Com um conjunto mínimo de rotas, percorre-se todos os possíveis links de comunicação entre nós obtendo suas respectivas RSSIs.

A terceira e proposta é uma metodologia para determinação de melhores rotas. Esta metodologia aborda de forma distinta quais as melhores rotas para aplicações de baixa e alta criticidade. Utilizando como critério de escolha, confiabilidade em aplicações de alta criticidade e menor latência em aplicações de baixa criticidade. Esta proposta interage duas propostas anteriores, inferindo a PER a partir de todas as RSSIs coletadas pelo protocolo.

Trabalhos futuros:

- Desenvolver um protocolo de roteamento para as melhores rotas que forem determinadas.
 - Estudar a possibilidade de efetuar este roteamento a partir de cálculos matemáticos, semelhante aos utilizados no protocolo de coleta.
- Refinar a estimativa da PER pela RSSI:
 - Realizar mais testes com diferentes intensidades de sinal, semelhantes aos já realizados.
 - Estudar o impacto da LQI na relação RSSI e PER

- Realizar um tratamento estatístico mais refinado na estimativa.
- Desenvolver rotas alternativas ao protocolo de coleta de RSSI, mantendo suas características de funcionamento principais e possibilitando sua aplicação em redes maiores, em área.
- Trabalhar a metodologia de criação de melhores rotas:
 - Aferir o impacto do número de saltos na latência.
 - Elaborar métodos alternativos de desenvolvimento de rotas.

Referências

AKYILDIZ, Ian F; VURAN, Mehmet C. et al. Wireless sensor networks: a survey. *Computer networks*, v. 38, n. 4, 2011.

[ANATEL] <http://www.anatel.gov.br/legislacao/resolucoes/2008/104-resolucao-506>. Acessado em Janeiro, 2016.

[BE900] Datasheet modulo BE900. Disponível em <http://radioit.com.br/wp-content/uploads/2015/06/BE900-Datasheet-v1.5.pdf>. Acessado em Outubro, 2015.

BRANQUINHO, Omar. Plataforma Radiuino para Estudos em Redes de Sensores Sem Fio, Disponível em: <http://www.radiuino.cc>, Acessado em Outubro, 2015.

[IEEE] Standard for Information Technology – Telecommunications and Information Exchange Between Systems – Local and Metropolitan Area Networks – Specific Requirement Part 15.4: Wireless Medium Access Control (MAC) And Physical Layer (PHY) Specifications For Low-Rate Wireless Personal Area Networks (WPANS). IEEE Std. 802.15.4a-2007 (Amendment to IEEE Std. 802.15.4-2006), pp. 1–203, 2007.

[Industry 4.0] The Engineer. Industry 4.0: the next industrial revolution. Disponível em: <<http://www.theengineer.co.uk/manufacturing/automation/industry-40-the-next-industrial-revolution/1016696.article>>. Acesso em junho 2015.

[ISA100] ISA 100. Disponível em: <https://www.isa.org/isa100/>. Acessado em Março, 2016

[ISA100b] ISA Setting the Standard for Automation. Control Over Wireless Current Applications and Future Opportunities. Disponível em: <http://www.nivis.com/resources/WCI_Auto_Week_2012_%20Paper_v18Sep.pdf>. Acessado em Agosto, 2015.

KARL, Holger; WILLING, Andreas. *Protocols and Architectures for Wireless Sensor Network*. John Wiley & Sons Ltd. 2005.

LATHI, B. P. *Modern dDigital and Analog Communication System*. 3rd. ed. New York: Oxford University Press, !988.

MADEIRA, J. F. L.; BRANQUINHO, O. C.. Implementation of Network Management with SNMP for Wireless Sensor Networks on Internet of Things. In: XXXI Simpósio Brasileiro de Telecomunicações, 2013, Fortaleza. Anais XXXI Simpósio Brasileiro de Telecomunicações, 2013.

OLIVEIRA, Rodolfo F.; BRANQUINHO, Omar C.; CHAVES, Pedro; LINO, Fernando. *Protocolo para gerenciamento de rotas em Redes de Sensores sem Fio*. 2016.

PEREIRA, V. Q.; BRANQUINHO, O. C.; REAL, C. R.; CHAVES, P.; LINO, F.. Proposta de Índices para Avaliação de Instabilidade de Sinal em RSSF Operando em Ambiente Industrial. In: XXXIII SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES, 2015, Juiz de Fora. ANAIS DO XXXIII SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES, 2015.

PRESSMAN, Roger S. Engenharia de Software. Tradução: Rosângela Delloso Penteado. 2006.

SRIVASTAVA, Vineet; MOTANI, Mehul. Cross-layer design: a survey and the road ahead. Communications Magazine, IEEE, v. 43, n. 12, p. 112-119, 2005.

STALLINGS, William. SNMP, SNMPv2, SNMPv3, and RMON 1 and 2 (3rd Edition). 1999.

[ZigBee]. ZigBee Alliance. Disponível em <<http://www.zigbee.org/>>. Acessado em maio, 2016

[WirelessHART] WirelessHART communications protocol. <http://www.hartcomm2.org/index.html>. Acessado em maio, 2016.

[6LoWPAN] IPv6 over low power WPAN working group. Disponível em <http://tools.ietf.org/wg/6lowpan/>. Acessado em Novembro, 2015.

Anexo A – Firmware Radiuino – Aferição da PER – Transmissor

Versão utilizada: RadiuinoEzRF_Timer

Adaptação do temporizador:

De

```
Timer1.initialize(1000000);
```

Para

```
Timer1.initialize(200000);
```

Adaptação da função Timer1:

```
void IntTimer1()
```

```
{
```

```
//O programa entrará nessa função a cada 0,2 segundo e transmitirá o pacote para a base.
```

```
pkt->IO0[0] = pkt->IO0[0] + 1;
```

```
if(pkt->IO0[0] == 0)
```

```
{
```

```
    pkt->IO0[1] = pkt->IO0[1] + 1;
```

```
if(pkt->IO0[1] == 0)
```

```
{
```

```
    pkt->IO0[2] = pkt->IO0[2] + 1;
```

```
if(pkt->IO0[2] == 0)
```

```
{
```

```
    pkt->IO1[0] = pkt->IO1[0] + 1;
```

```
if(pkt->IO1[0] == 0)
```

```
{
```

```
    pkt->IO1[1] = pkt->IO1[1] + 1;
```

```
if(pkt->IO1[1] == 0)
```

```
{
```

```
    pkt->IO1[2] = pkt->IO1[2] + 1;
```

```
}
```

```
}  
}  
}  
}  
App.send(TxPacket); //Transmite o pacote.  
}
```

Anexo B – Firmware Radiuino – Aferição da PER – Receptor

Versão utilizada: RadiuinoEzRF_Base

Adaptação na função RX:

```
void RF_ReceivedPacket()
{
    pkt->IO2[0] = pkt->IO2[0] + 1;
    if(pkt->IO2[0] == 0)
    {
        pkt->IO2[1] = pkt->IO2[1] + 1;
        if(pkt->IO2[1] == 0)
        {
            pkt->IO2[2] = pkt->IO2[2] + 1;
            if(pkt->IO2[2] == 0)
            {
                pkt->IO3[0] = pkt->IO3[0] + 1;
                if(pkt->IO3[0] == 0)
                {
                    pkt->IO3[1] = pkt->IO3[1] + 1;
                    if(pkt->IO3[1] == 0)
                    {
                        pkt->IO3[2] = pkt->IO3[2] + 1;
                    }
                }
            }
        }
    }

    /* Reenvia o pacote recebido pelo RF pela Serial.      */
    Phy.sendSerial(RxPacket);
}
```

Anexo C – Firmware Radiuino – Protocolo para Coleta de RSSI

Versão utilizada: RadiuinoStack_Pooling

Adaptação da camada de rede

//Função:

void NET::receive(packet * pkt)

```
{
  if(pkt->NetHdr[0] == Net.my_addr || pkt->NetHdr[0] == Net.my_addr + pkt->NetHdr[3])
  {
    /* Envia para a camada superior */
    Transp.receive(pkt);
  }
  else
  {
    return;
  }
}
```

//Função:

void NET::send(packet * pkt)

```
{
  //Coloca o endereço de destino no pacote.
  //(O destino é o endereço da placa que enviou o pacote anteriormente).
  pkt->NetHdr[0] = Net.my_addr + pkt->NetHdr[1];
  if(pkt->NetHdr[0] >> pkt->NetHdr[3])
  {
    pkt->NetHdr[0] = pkt->NetHdr[0] - pkt->NetHdr[3];
  }

  if(pkt->NetHdr[0] == Net.my_addr || pkt->NetHdr[0] == Net.my_addr + pkt->NetHdr[3])
```

```

{
    pkt->NetHdr[0] = 0;
}

pkt->NetHdr[1] = pkt->NetHdr[1]+1;

Mac.send(pkt);
}

```

Adaptação da Camada de Aplicação

//Função:

void APP::receive(packet * pkt)

```

{
    switch (pkt->NetHdr[1])
    {
        case 7:
            pkt->IO0[0] = pkt->PhyHdr[0];
            pkt->IO0[1] = pkt->PhyHdr[1];
            pkt->IO0[2] = Net.my_addr;
            break;
        case 8:
            pkt->IO1[0] = pkt->PhyHdr[0];
            pkt->IO1[1] = pkt->PhyHdr[1];
            pkt->IO1[2] = Net.my_addr;
            break;
        case 9:
            pkt->IO2[0] = pkt->PhyHdr[0];
            pkt->IO2[1] = pkt->PhyHdr[1];
            pkt->IO2[2] = Net.my_addr;
            break;
        case 10:
            pkt->IO3[0] = pkt->PhyHdr[0];
            pkt->IO3[1] = pkt->PhyHdr[1];

```

```
pkt->IO3[2] = Net.my_addr;
break;
case 11:
pkt->IO4[0] = pkt->PhyHdr[0];
pkt->IO4[1] = pkt->PhyHdr[1];
pkt->IO4[2] = Net.my_addr;
    break;
case 12:
pkt->IO5[0] = pkt->PhyHdr[0];
pkt->IO5[1] = pkt->PhyHdr[1];
pkt->IO5[2] = Net.my_addr;
    break;
case 1:
pkt->AD0[0] = pkt->PhyHdr[0];
pkt->AD0[1] = pkt->PhyHdr[1];
pkt->AD0[2] = Net.my_addr;
    break;
case 2:
pkt->AD1[0] = pkt->PhyHdr[0];
pkt->AD1[1] = pkt->PhyHdr[1];
pkt->AD1[2] = Net.my_addr;
    break;
case 3:
pkt->AD2[0] = pkt->PhyHdr[0];
pkt->AD2[1] = pkt->PhyHdr[1];
pkt->AD2[2] = Net.my_addr;
    break;
case 4:
pkt->AD3[0] = pkt->PhyHdr[0];
pkt->AD3[1] = pkt->PhyHdr[1];
pkt->AD3[2] = Net.my_addr;
```

```
    break;

    case 5:

    pkt->AD4[0] = pkt->PhyHdr[0];
    pkt->AD4[1] = pkt->PhyHdr[1];
    pkt->AD4[2] = Net.my_addr;

    break;

    case 6:

    pkt->AD5[0] = pkt->PhyHdr[0];
    pkt->AD5[1] = pkt->PhyHdr[1];
    pkt->AD5[2] = Net.my_addr;

    break;

    }

    // TRANSMITE PACOTE

    App.send(*pkt);

    return;

}
```

Anexo D – Software Python – Aferição da PER

```

# TESTE 15 SENSORES

# PROGRAMA para medição de temperatura e pulsação

import serial

import math

import time

import struct

from time import localtime, strftime

#=====CONFIGURA SERIAL (USB)=====

# para COM# o número que se coloca é n-1 no primeiro parâmetro. Ex COM9 valor 8

n_serial = raw_input("Digite o número da serial = ") #seta a serial

n_serial1 = int(n_serial) - 1

ser = serial.Serial(n_serial1, 9600, timeout=10,parity=serial.PARITY_NONE) # seta valores da serial

#===== VARIÁVEIS UTILIZADAS =====

CONTADOR = 0 # conta o número de pollings realizados

TEMP1 = TEMP2 = TEMP3 = UMID1 = UMID2 = UMID3 = RSSId1 = RSSId2 = RSSId3 =
RSSId4 = RSSId5 = RSSId6 = RSSId7 = RSSId8 = RSSId9 =
RSSId10 = RSSId11 = RSSId12 = RSSId13 = RSSId14 =
RSSId15 = RSSId16 = RSSId17 = RSSId18 = RSSId19 =
RSSId20 = RSSId21 = RSSId22 = RSSId23 = RSSId24 =
RSSId25 = RSSId26 = RSSId27 = RSSId28 = RSSId29 =
RSSId30 = RSSId31 = RSSId32 = RSSId33 = RSSId34 =
RSSId35 = RSSId36 = RSSId37 = RSSId38 = RSSId39 =
RSSId40 = RSSId41 = RSSId42 = RSSId43 = RSSId44 =
RSSId45 = RSSId46 = RSSId47 = RSSId48 = RSSId49 =
RSSId50 = RSSId51 = RSSId52 = RSSId53 = RSSId54 =
RSSId55 = RSSId56 = RSSId57 = RSSId58 = RSSId59 =
RSSId60 = RSSId61 = RSSId62 = RSSId63 = RSSId64 =
RSSId65 = RSSId66 = RSSId67 = RSSId68 = RSSId69 =
RSSId70 = RSSId71 = RSSId72 = RSSId73 = RSSId74 =
RSSId75 = RSSId76 = RSSId77 = RSSId78 = RSSId79 =
RSSId80 = RSSId81 = RSSId82 = RSSId83 = RSSId84 =
RSSId85 = RSSId86 = RSSId87 = RSSId88 = RSSId89 =
RSSId90 = RSSId91 = RSSId92 = RSSId93 = RSSId94 =
RSSId95 = RSSId96 = RSSId97 = RSSId98 = RSSId99 =
RSSId100 = RSSId101 = RSSId102 = RSSId103 = RSSId104 =
RSSId105 = RSSId106 = RSSId107 = RSSId108 = RSSId109 =
RSSId110 = RSSId111 = RSSId112 = RSSId113 = RSSId114 =
RSSId115 = RSSId116 = RSSId117 = RSSId118 = RSSId119 =
RSSId120 = RSSId121 = RSSId122 = RSSId123 = RSSId124 =
RSSId125 = RSSId126 = RSSId127 = RSSId128 = RSSId129 =
RSSId130 = RSSId131 = RSSId132 = RSSId133 = RSSId134 =
RSSId135 = RSSId136 = RSSId137 = RSSId138 = RSSId139 =
RSSId140 = RSSId141 = RSSId142 = RSSId143 = RSSId144 =
RSSId145 = RSSId146 = RSSId147 = RSSId148 = RSSId149 =
RSSId150 = RSSId151 = RSSId152 = RSSId153 = RSSId154 =
RSSId155 = RSSId156 = RSSId157 = RSSId158 = RSSId159 =
RSSId160 = RSSId161 = RSSId162 = RSSId163 = RSSId164 =
RSSId165 = RSSId166 = RSSId167 = RSSId168 = RSSId169 =
RSSId170 = RSSId171 = RSSId172 = RSSId173 = RSSId174 =
RSSId175 = RSSId176 = RSSId177 = RSSId178 = RSSId179 =
RSSId180 = RSSId181 = RSSId182 = RSSId183 = RSSId184 =
RSSId185 = RSSId186 = RSSId187 = RSSId188 = RSSId189 =
RSSId190 = RSSId191 = RSSId192 = RSSId193 = RSSId194 =
RSSId195 = RSSId196 = RSSId197 = RSSId198 = RSSId199 =
RSSId200 = RSSId201 = RSSId202 = RSSId203 = RSSId204 =
RSSId205 = RSSId206 = RSSId207 = RSSId208 = RSSId209 =
RSSId210 = RSSId211 = RSSId212 = RSSId213 = RSSId214 =
RSSId215 = RSSId216 = RSSId217 = RSSId218 = RSSId219 =
RSSId220 = RSSId221 = RSSId222 = RSSId223 = RSSId224 =
RSSId225 = RSSId226 = RSSId227 = RSSId228 = RSSId229 =
RSSId230 = RSSId231 = RSSId232 = RSSId233 = RSSId234 =
RSSId235 = RSSId236 = RSSId237 = RSSId238 = RSSId239 =
RSSId240 = RSSId241 = RSSId242 = RSSId243 = RSSId244 =
RSSId245 = RSSId246 = RSSId247 = RSSId248 = RSSId249 =
RSSId250 = RSSId251 = RSSId252 = RSSId253 = RSSId254 =
RSSId255 = RSSId256 = RSSId257 = RSSId258 = RSSId259 =
RSSId260 = RSSId261 = RSSId262 = RSSId263 = RSSId264 =
RSSId265 = RSSId266 = RSSId267 = RSSId268 = RSSId269 =
RSSId270 = RSSId271 = RSSId272 = RSSId273 = RSSId274 =
RSSId275 = RSSId276 = RSSId277 = RSSId278 = RSSId279 =
RSSId280 = RSSId281 = RSSId282 = RSSId283 = RSSId284 =
RSSId285 = RSSId286 = RSSId287 = RSSId288 = RSSId289 =
RSSId290 = RSSId291 = RSSId292 = RSSId293 = RSSId294 =
RSSId295 = RSSId296 = RSSId297 = RSSId298 = RSSId299 =
RSSId300 = RSSId301 = RSSId302 = RSSId303 = RSSId304 =
RSSId305 = RSSId306 = RSSId307 = RSSId308 = RSSId309 =
RSSId310 = RSSId311 = RSSId312 = RSSId313 = RSSId314 =
RSSId315 = RSSId316 = RSSId317 = RSSId318 = RSSId319 =
RSSId320 = RSSId321 = RSSId322 = RSSId323 = RSSId324 =
RSSId325 = RSSId326 = RSSId327 = RSSId328 = RSSId329 =
RSSId330 = RSSId331 = RSSId332 = RSSId333 = RSSId334 =
RSSId335 = RSSId336 = RSSId337 = RSSId338 = RSSId339 =
RSSId340 = RSSId341 = RSSId342 = RSSId343 = RSSId344 =
RSSId345 = RSSId346 = RSSId347 = RSSId348 = RSSId349 =
RSSId350 = RSSId351 = RSSId352 = RSSId353 = RSSId354 =
RSSId355 = RSSId356 = RSSId357 = RSSId358 = RSSId359 =
RSSId360 = RSSId361 = RSSId362 = RSSId363 = RSSId364 =
RSSId365 = RSSId366 = RSSId367 = RSSId368 = RSSId369 =
RSSId370 = RSSId371 = RSSId372 = RSSId373 = RSSId374 =
RSSId375 = RSSId376 = RSSId377 = RSSId378 = RSSId379 =
RSSId380 = RSSId381 = RSSId382 = RSSId383 = RSSId384 =
RSSId385 = RSSId386 = RSSId387 = RSSId388 = RSSId389 =
RSSId390 = RSSId391 = RSSId392 = RSSId393 = RSSId394 =
RSSId395 = RSSId396 = RSSId397 = RSSId398 = RSSId399 =
RSSId400 = RSSId401 = RSSId402 = RSSId403 = RSSId404 =
RSSId405 = RSSId406 = RSSId407 = RSSId408 = RSSId409 =
RSSId410 = RSSId411 = RSSId412 = RSSId413 = RSSId414 =
RSSId415 = RSSId416 = RSSId417 = RSSId418 = RSSId419 =
RSSId420 = RSSId421 = RSSId422 = RSSId423 = RSSId424 =
RSSId425 = RSSId426 = RSSId427 = RSSId428 = RSSId429 =
RSSId430 = RSSId431 = RSSId432 = RSSId433 = RSSId434 =
RSSId435 = RSSId436 = RSSId437 = RSSId438 = RSSId439 =
RSSId440 = RSSId441 = RSSId442 = RSSId443 = RSSId444 =
RSSId445 = RSSId446 = RSSId447 = RSSId448 = RSSId449 =
RSSId450 = RSSId451 = RSSId452 = RSSId453 = RSSId454 =
RSSId455 = RSSId456 = RSSId457 = RSSId458 = RSSId459 =
RSSId460 = RSSId461 = RSSId462 = RSSId463 = RSSId464 =
RSSId465 = RSSId466 = RSSId467 = RSSId468 = RSSId469 =
RSSId470 = RSSId471 = RSSId472 = RSSId473 = RSSId474 =
RSSId475 = RSSId476 = RSSId477 = RSSId478 = RSSId479 =
RSSId480 = RSSId481 = RSSId482 = RSSId483 = RSSId484 =
RSSId485 = RSSId486 = RSSId487 = RSSId488 = RSSId489 =
RSSId490 = RSSId491 = RSSId492 = RSSId493 = RSSId494 =
RSSId495 = RSSId496 = RSSId497 = RSSId498 = RSSId499 =
RSSId500 = RSSId501 = RSSId502 = RSSId503 = RSSId504 =
RSSId505 = RSSId506 = RSSId507 = RSSId508 = RSSId509 =
RSSId510 = RSSId511 = RSSId512 = RSSId513 = RSSId514 =
RSSId515 = RSSId516 = RSSId517 = RSSId518 = RSSId519 =
RSSId520 = RSSId521 = RSSId522 = RSSId523 = RSSId524 =
RSSId525 = RSSId526 = RSSId527 = RSSId528 = RSSId529 =
RSSId530 = RSSId531 = RSSId532 = RSSId533 = RSSId534 =
RSSId535 = RSSId536 = RSSId537 = RSSId538 = RSSId539 =
RSSId540 = RSSId541 = RSSId542 = RSSId543 = RSSId544 =
RSSId545 = RSSId546 = RSSId547 = RSSId548 = RSSId549 =
RSSId550 = RSSId551 = RSSId552 = RSSId553 = RSSId554 =
RSSId555 = RSSId556 = RSSId557 = RSSId558 = RSSId559 =
RSSId560 = RSSId561 = RSSId562 = RSSId563 = RSSId564 =
RSSId565 = RSSId566 = RSSId567 = RSSId568 = RSSId569 =
RSSId570 = RSSId571 = RSSId572 = RSSId573 = RSSId574 =
RSSId575 = RSSId576 = RSSId577 = RSSId578 = RSSId579 =
RSSId580 = RSSId581 = RSSId582 = RSSId583 = RSSId584 =
RSSId585 = RSSId586 = RSSId587 = RSSId588 = RSSId589 =
RSSId590 = RSSId591 = RSSId592 = RSSId593 = RSSId594 =
RSSId595 = RSSId596 = RSSId597 = RSSId598 = RSSId599 =
RSSId600 = RSSId601 = RSSId602 = RSSId603 = RSSId604 =
RSSId605 = RSSId606 = RSSId607 = RSSId608 = RSSId609 =
RSSId610 = RSSId611 = RSSId612 = RSSId613 = RSSId614 =
RSSId615 = RSSId616 = RSSId617 = RSSId618 = RSSId619 =
RSSId620 = RSSId621 = RSSId622 = RSSId623 = RSSId624 =
RSSId625 = RSSId626 = RSSId627 = RSSId628 = RSSId629 =
RSSId630 = RSSId631 = RSSId632 = RSSId633 = RSSId634 =
RSSId635 = RSSId636 = RSSId637 = RSSId638 = RSSId639 =
RSSId640 = RSSId641 = RSSId642 = RSSId643 = RSSId644 =
RSSId645 = RSSId646 = RSSId647 = RSSId648 = RSSId649 =
RSSId650 = RSSId651 = RSSId652 = RSSId653 = RSSId654 =
RSSId655 = RSSId656 = RSSId657 = RSSId658 = RSSId659 =
RSSId660 = RSSId661 = RSSId662 = RSSId663 = RSSId664 =
RSSId665 = RSSId666 = RSSId667 = RSSId668 = RSSId669 =
RSSId670 = RSSId671 = RSSId672 = RSSId673 = RSSId674 =
RSSId675 = RSSId676 = RSSId677 = RSSId678 = RSSId679 =
RSSId680 = RSSId681 = RSSId682 = RSSId683 = RSSId684 =
RSSId685 = RSSId686 = RSSId687 = RSSId688 = RSSId689 =
RSSId690 = RSSId691 = RSSId692 = RSSId693 = RSSId694 =
RSSId695 = RSSId696 = RSSId697 = RSSId698 = RSSId699 =
RSSId700 = RSSId701 = RSSId702 = RSSId703 = RSSId704 =
RSSId705 = RSSId706 = RSSId707 = RSSId708 = RSSId709 =
RSSId710 = RSSId711 = RSSId712 = RSSId713 = RSSId714 =
RSSId715 = RSSId716 = RSSId717 = RSSId718 = RSSId719 =
RSSId720 = RSSId721 = RSSId722 = RSSId723 = RSSId724 =
RSSId725 = RSSId726 = RSSId727 = RSSId728 = RSSId729 =
RSSId730 = RSSId731 = RSSId732 = RSSId733 = RSSId734 =
RSSId735 = RSSId736 = RSSId737 = RSSId738 = RSSId739 =
RSSId740 = RSSId741 = RSSId742 = RSSId743 = RSSId744 =
RSSId745 = RSSId746 = RSSId747 = RSSId748 = RSSId749 =
RSSId750 = RSSId751 = RSSId752 = RSSId753 = RSSId754 =
RSSId755 = RSSId756 = RSSId757 = RSSId758 = RSSId759 =
RSSId760 = RSSId761 = RSSId762 = RSSId763 = RSSId764 =
RSSId765 = RSSId766 = RSSId767 = RSSId768 = RSSId769 =
RSSId770 = RSSId771 = RSSId772 = RSSId773 = RSSId774 =
RSSId775 = RSSId776 = RSSId777 = RSSId778 = RSSId779 =
RSSId780 = RSSId781 = RSSId782 = RSSId783 = RSSId784 =
RSSId785 = RSSId786 = RSSId787 = RSSId788 = RSSId789 =
RSSId790 = RSSId791 = RSSId792 = RSSId793 = RSSId794 =
RSSId795 = RSSId796 = RSSId797 = RSSId798 = RSSId799 =
RSSId800 = RSSId801 = RSSId802 = RSSId803 = RSSId804 =
RSSId805 = RSSId806 = RSSId807 = RSSId808 = RSSId809 =
RSSId810 = RSSId811 = RSSId812 = RSSId813 = RSSId814 =
RSSId815 = RSSId816 = RSSId817 = RSSId818 = RSSId819 =
RSSId820 = RSSId821 = RSSId822 = RSSId823 = RSSId824 =
RSSId825 = RSSId826 = RSSId827 = RSSId828 = RSSId829 =
RSSId830 = RSSId831 = RSSId832 = RSSId833 = RSSId834 =
RSSId835 = RSSId836 = RSSId837 = RSSId838 = RSSId839 =
RSSId840 = RSSId841 = RSSId842 = RSSId843 = RSSId844 =
RSSId845 = RSSId846 = RSSId847 = RSSId848 = RSSId849 =
RSSId850 = RSSId851 = RSSId852 = RSSId853 = RSSId854 =
RSSId855 = RSSId856 = RSSId857 = RSSId858 = RSSId859 =
RSSId860 = RSSId861 = RSSId862 = RSSId863 = RSSId864 =
RSSId865 = RSSId866 = RSSId867 = RSSId868 = RSSId869 =
RSSId870 = RSSId871 = RSSId872 = RSSId873 = RSSId874 =
RSSId875 = RSSId876 = RSSId877 = RSSId878 = RSSId879 =
RSSId880 = RSSId881 = RSSId882 = RSSId883 = RSSId884 =
RSSId885 = RSSId886 = RSSId887 = RSSId888 = RSSId889 =
RSSId890 = RSSId891 = RSSId892 = RSSId893 = RSSId894 =
RSSId895 = RSSId896 = RSSId897 = RSSId898 = RSSId899 =
RSSId900 = RSSId901 = RSSId902 = RSSId903 = RSSId904 =
RSSId905 = RSSId906 = RSSId907 = RSSId908 = RSSId909 =
RSSId910 = RSSId911 = RSSId912 = RSSId913 = RSSId914 =
RSSId915 = RSSId916 = RSSId917 = RSSId918 = RSSId919 =
RSSId920 = RSSId921 = RSSId922 = RSSId923 = RSSId924 =
RSSId925 = RSSId926 = RSSId927 = RSSId928 = RSSId929 =
RSSId930 = RSSId931 = RSSId932 = RSSId933 = RSSId934 =
RSSId935 = RSSId936 = RSSId937 = RSSId938 = RSSId939 =
RSSId940 = RSSId941 = RSSId942 = RSSId943 = RSSId944 =
RSSId945 = RSSId946 = RSSId947 = RSSId948 = RSSId949 =
RSSId950 = RSSId951 = RSSId952 = RSSId953 = RSSId954 =
RSSId955 = RSSId956 = RSSId957 = RSSId958 = RSSId959 =
RSSId960 = RSSId961 = RSSId962 = RSSId963 = RSSId964 =
RSSId965 = RSSId966 = RSSId967 = RSSId968 = RSSId969 =
RSSId970 = RSSId971 = RSSId972 = RSSId973 = RSSId974 =
RSSId975 = RSSId976 = RSSId977 = RSSId978 = RSSId979 =
RSSId980 = RSSId981 = RSSId982 = RSSId983 = RSSId984 =
RSSId985 = RSSId986 = RSSId987 = RSSId988 = RSSId989 =
RSSId990 = RSSId991 = RSSId992 = RSSId993 = RSSId994 =
RSSId995 = RSSId996 = RSSId997 = RSSId998 = RSSId999 =

#===== CRIAÇÃO DOS PACOTES PARA CADA SENSOR =====

Pacote = {}

# Cria Pacote de 52 bytes com valor zero em todas as posições.

for i in range(0,52): # faz um array com 52 bytes

    Pacote[i] = 0

#===== LOOP DE EXECUÇÃO =====

while True:

    try:

        # Imprime na tela o menu de opções

        print 'Escolha um comandos abaixo e depois enter'

        print '1 - Realiza medidas:'

```

```

print 's - Para sair:'

# Entrada da opção
Opcao = raw_input('Entre com a Opção = ')

# Limpa o buffer da serial
ser.flushInput()

# Verifica se é para encerrar a execução
if Opcao == "s" or Opcao == "S":# caso o caracter digitado for s
    ser.close() # fecha a porta COM

    print 'Fim da Execução' # escreve na tela

    break

# Execução das medidas
if Opcao == "1":
    # Identificação da base
    #ID_base = raw_input('ID_base = ')

    #ID_base = 0

    # Identificação do sensor a ser acessado
    #ID_sensor1 = raw_input('ID do sensor 1 = ')

    #ID_sensor1 = 2

    # Coloca nos pacotes para cada sensor a identificação da base e do sensor
    #Pacote[8] = int(ID_sensor1)

    #Pacote[10] = int(ID_base)

    # Entra com o número de medidas que devem ser realizadas
    num_medidas = raw_input('Entre com o número de medidas = ')

#===== CRIA ARQUIVO PARA GRAVAR O LOG =====
FileLOG = strftime("LOG_%Y_%m_%d_%H-%M-%S.txt")
print "Arquivo de log: %s" % FileLOG

LOG = open(FileLOG, 'w') # Abre o arquivo filename1 com permissão de escrita

#=====FOR PARA REALIZAÇÃO DAS MEDIDAS
for j in range(0,int(num_medidas)): # for para realização das w medidas

```

```

ser.flushInput() # Limpa o buffer da serial

#print '===== Número de Pollings =',CONTADOR,'=====

#=====COMUNICAÇÃO DA BASE COM O SENSOR 1

# Transmissão do pacote de leitura

#for k in range(0,52):

    #TXbyte = chr(Pacote[k])

    #ser.write(TXbyte)

# Aguarda a resposta do sensor

#time.sleep(int(TEMPO1))

line = ser.read(52) # faz a leitura de 52 bytes do buffer que recebe da serial pela COM

if len(line) != 52:

    erro_rx = erro_rx + 1

    print 'Erro RX falha 52 bytes = ',erro_rx

if len(line) == 52:

    rssid1 = ord(line[0]) # RSSI_DownLink

    rssi1 = ord(line[2]) # RSSI_UpLink

    LQI = ord(line[3])

    #RSSI Downlink - potência recebida pelo sensor

    if rssid1 > 128:

        RSSId1=((rssid1-256)/2.0)-74

    else:

        RSSId1=(rssid1/2.0)-74

    #RSSI Uplink - potência recebida pela base

    if rssi1 > 128:

        RSSId1=((rssi1-256)/2.0)-74

    else:

        RSSId1=(rssi1/2.0)-74

count = ord(line[10]) # contador de pacotes enviados pelo sensor

# Contadores envio

e1 = ord(line[34])

```

```

e2 = ord(line[35])
e3 = ord(line[36])
e4 = ord(line[37])
e5 = ord(line[38])
e6 = ord(line[39])

# contadores recepção
r1 = ord(line[40])
r2 = ord(line[41])
r3 = ord(line[42])
r4 = ord(line[43])
r5 = ord(line[44])
r6 = ord(line[45])

ENV =
e1+(e2*256)+(e3*256*256)+(e4*256*256*256)+(e5*256*256*256*256)+(e6*256*256*256*256*256)

REC =
r1+(r2*256)+(r3*256*256)+(r4*256*256*256)+(r5*256*256*256*256)+(r6*256*256*256*256*256)

PER = ENV-REC

#Temperatura = (ad1h * 256 + ad1l)/100.00

#Temperatura = ord(line[12])

print 'PER= ', PER, ' ENVIADO= ',ENV, ' RECEBIDO= ',REC, ' RSSI= ',RSSIu1, ' LQI= ',LQI

print >>LOG,time.asctime(),'=PER= ', PER, '=ENVIADO= ',ENV, '=RECEBIDO= ',REC, '=RSSI= ',RSSIu1,
=LQI= ',LQI

#time.sleep(int(TEMPO3))

LOG.close()

except KeyboardInterrupt:

ser.close()

break

```

Anexo E – Software Phyton – Coleta de RSSIs

```

# -*- coding: utf-8 -*-

#Desenvolvedor e Programador: Rodolfo Francisco de Oliveira

#Importar as bibliotecas necessarias

import serial

import time

from time import localtime, strftime

from datetime import datetime, date

import struct

#Identifica a porta serial a ser utilizada pelo sistema

#porta_serial = '/dev/ttyUSB0'

n_serial = raw_input("Digite o número da serial:") #seta a serial

n_serial1 = int(n_serial) - 1

ser = serial.Serial(n_serial1, 9600, timeout=0.5,parity=serial.PARITY_NONE) # seta valores da serial

#Identifica a quantidade de nós sensores na rede.

quantidade_nos = int(raw_input('Digite a quantidade de sensores da rede:'))

#Identifica a quantidade de vezes que o algoritmo irá rodar

quantidade_vezes = int(raw_input('Digite a quantidade de ciclos:'))

#Identifica o id (endereço) da base

id_base = 0

#Cria vetor de respostas de 36 bytes e preenche o mesmo com "0"

resposta = {}

for i in range(1,37):

    resposta[i] = 0

#Cria objeto para porta serial

#try:

# serialPort = serial.Serial(porta_serial, 9600, timeout=1, parity=serial.PARITY_NONE)

#except Exception:

#Imprime mensagem de erro e encerra o programa

# print "Erro! Não foi possível abrir a porta" , porta_serial, ". Corrija o problema"

# quit()

```

```

#n_serial = raw_input("Digite o número da serial:") #seta a serial
#n_serial1 = int(n_serial) - 1

#ser = serial.Serial(n_serial1, 9600, timeout=0.5,parity=serial.PARITY_NONE) # seta valores da serial

#Cria arquivo para armazenar os logs
nome_arquivo = strftime("Medida_%Y_%m_%d_%H-%M-%S.csv")
print "Arquivo de log: %s" % nome_arquivo
arquivo = open(nome_arquivo, 'w')

#Repetição principal do aplicativo
for x in range(1,quantidade_vezes+1):
    print 'Iteração nº',x

    #Começa o procedimento para descoberta das rotas dos nós. Vai percorrer todos os nós, a principio.

    #Caso algum nó já tenha uma rota estabelecida (Cadastrada como 'S'), ele será pulado.
    for i in range(1,quantidade_nos+1):

        #Cria um pacote de 52 bytes vazio para ser enviado
        pacoteSX = {}

        for j in range(0,52):
            pacoteSX[j] = 0

        #Começa a configurar o pacote que será enviado.

        #Configura o byte de posição 10 (referente ao endereço de origem) com o endereço da base
        pacoteSX[10] = id_base

        #Configura o byte de posição 8 (referente ao endereço de destino) com o endereço do nó
        pacoteSX[8] = i

        #Configura o byte de posição 9 (numero de saltos)
        pacoteSX[9] = 1

        #Configura o byte de posição 11 (total de sensores)
        pacoteSX[11] = quantidade_nos

        #Limpa os buffers de entrada e saída da serial
        ser.flushInput()
        ser.flushOutput()

        #Transmite o pacote na serial
        for j in range(0,52):

```

```

ser.write(chr(pacoteSX[j]))

#Espera um tempo para que o pacote seja transmitido e recebido novamente
time.sleep(0.0)

#Le a serial em busca do pacote
pacoteRX = ser.read(52)

#Verifica se o pacote está integro e se é um retorno de rota
if len(pacoteRX) == 52:

    #Calculo dos RSSI de Download e Upload
    tempRSSIDownload = ord(pacoteRX[0])
    tempRSSIUpload = ord(pacoteRX[2])

    #Pega a LQI da Base
    lqiBase = ord(pacoteRX[1])

    #Calcula RSSI de Download e Upload
    if tempRSSIDownload > 128:
        rssiDownloadBase = ((tempRSSIDownload-256)/2.0)-74
    else:
        rssiDownloadBase = (tempRSSIDownload/2.0)-74
    if tempRSSIUpload > 128:
        rssiUploadBase = ((tempRSSIUpload-256)/2.0)-74
    else:
        rssiUploadBase = (tempRSSIUpload/2.0)-74

    #Contador para controlar a posição de escrita dos pacotes (w)
    w = 1

    #Contador para escrever no vetor de respostas
    contador = 2

    lqi = 0
    idNo = 0
    resposta[1] = 0

    for y in range(1,13):
        tempRSSI = ord(pacoteRX[15+w])
        if tempRSSI > 128:

```

```

    rssi = ((tempRSSI-256)/2.0)-74
else:
    rssi = (tempRSSI/2.0)-74
resposta[contador] = rssi
contador = contador + 1
lqi = ord(pacoteRX[16+w])
resposta[contador] = lqi
contador = contador + 1
idNo = ord(pacoteRX[17+w])
resposta[contador] = idNo
contador = contador + 1
w = w + 3
linhaImprime = ""
for k in range(1,contador):
    linhaImprime = linhaImprime + str(resposta[k]) + ';'
print >>arquivo,str(x),';',linhaImprime,';',str(rssiUploadBase),';',str(lqiBase),';'
else:
    print 'Erro na recepção!'

#Fecha arquivo
arquivo.close()

#Fecha a porta serial
ser.close()

```