

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE
CAMPINAS**

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

KARYNA SILVEIRA CARDOSO

**UM ESTUDO DE APLICAÇÃO DE BACKHAUL
HÍBRIDO DE ROF E RÁDIO EM RSSF**

CAMPINAS

2015

AVALIAÇÃO

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

Orientador: Prof. Dr. Omar Carvalho Branquinho

CAMPINAS

2015

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

t621.3851 Cardoso, Karyna Silveira.
C268e Um estudo de aplicação de Backhaul híbrido rof rádio em RSSF /
Karyna Silveira Cardoso. - Campinas: PUC-Campinas, 2015.
114p.

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

1. Redes de sensores sem fio. 2. Fibras óticas. 3. Sistemas de co-
municação sem fio. 4. Telecomunicações. I. Branquinho, Omar de Car-
valho. 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

KARYNA SILVEIRA CARDOSO

**UM ESTUDO DE APLICAÇÃO DE BACKHAUL HÍBRIDO
ROF RÁDIO 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: Qualidade de Serviço de Telemática. Orientador: Prof. Dr. Omar Carvalho Branquinho

Dissertação defendida e aprovada em 16 de junho de 2015 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



Prof. Dr. David Bianchini
Pontifícia Universidade Católica de Campinas



Prof. Dr. Fabiano Fruett
Universidade Estadual de Campinas

Campinas, 16 de maio de 2015.

Ao meu esposo Adolfo Blengini que não me deixou desistir nos momentos difíceis e me incentivou,
e ao meu orientador Omar Branquinho pela confiança e apoio em toda a jornada.

AGRADECIMENTOS

Ao Prof. Dr. Omar Carvalho Branquinho,
Orientador e incentivador do meu trabalho pelo apoio e pela amizade.

Ao analista Adolfo Blengini Neto,
Pela ajuda na programação em Python.

Ao Mestre Raphael Montali da ASSUMPÇÃO,
Pela ajuda na montagem da estrutura óptica do laboratório LPSira e pelo apoio nas dúvidas.

Ao engenheiro Alberto Imamura,
Pela ajuda na confecção das placas dos repetidores.

Ao engenheiro Vitor Queiroz,
Pela ajuda na montagem das células outdoor.

Ao mestre Tiago Pedroso,
Pela ajuda na programação.

A aluna de Graduação Maria Caroline ANDRADE,
Pela ajuda na coleta nos dados em bancada.

“O que sabemos é uma gota. O que ignoramos é um oceano.”

Isaac Newton.

RESUMO

CARDOSO, Karyna Silveira. Um estudo da aplicação de backhaul híbrido de RoF e rádio em RSSF. 2015. Folhas 114f. Dissertação (Mestrado em Gestão de Redes de Telecomunicações) – Pontifícia Universidade Católica de Campinas, Centro de Ciências Exatas, Ambientais e de Tecnologias, Programa de Pós-Graduação em Engenharia Elétrica, Campinas, 2015.

O presente trabalho propõe uma estratégia para acessar nós sensores de uma rede de sensores sem fio (RSSF) aglutinados em clusters que se unem em células através de uma topologia híbrida de rádio e/ou fibra. Nesta topologia, a comunicação entre as redes de sensores sem fio e o repetidor será via rádio, enquanto o *backhaul* de comunicação entre a base e os repetidores será híbrido, podendo utilizar rádio ou fibra. Portanto, a proposta abrange uma topologia flexível, utilizando tanto rádio quanto fibra no *backhaul* e acesso aos nós sensores via uma rede de sensores sem fio. Para demonstrar a proposta foram feitos testes em uma bancada de emulação de canal utilizando *backhaul* de fibra entre a base e um elemento repetidor. Estes testes foram realizados na faixa de 915 MHz, na modulação FSK variando a sua taxa de transmissão, para avaliar a área de cobertura para cada taxa. Como resultados foram obtidas curvas que mostram que para uma RSSI fixa, quanto maior a taxa de transmissão pior será a taxa de erros de bits (BER). E quanto maior o fator de atenuação do ambiente menor será a área de cobertura da rede em todas as taxas testadas. Logo quanto menor a potência recepção, maior a distância entre os nós sensores. Também foram realizados testes em um ambiente real para avaliar os protocolos implementados. Estes testes consistiam no acesso dos nós sensores dos clusters através de múltiplos saltos e medição dos valores de potência de recepção de cada nó sensor e sua taxa de perda de pacotes (PER). Através dos testes realizados foi comprovado o funcionamento da proposta implementada e a topologia flexível proposta.

Palavras-chave: Rede de sensores sem fio. Fibra óptica. Múltiplos saltos. Topologia híbrida.

ABSTRACT

CARDOSO, Karyna Silveira. *Study the application of hybrid RoF backhaul and radio in wireless sensor networks*. 2015. Folhas 114f. *Dissertation (Master of Engineering)* – Pontifícia Universidade Católica de Campinas, Centro de Ciências Exatas, Ambientais e de Tecnologias, Programa de Pós-Graduação em Engenharia Elétrica, Campinas, 2015.

This study proposes a strategy to access sensor nodes of a wireless sensor network clumped together in clusters that come together in cells through a hybrid radio topology and / or fiber. In this topology, communication between wireless sensor networks and the repeater is via radio, while the backhaul communication between the base and repeaters will be hybrid and can use radio or fiber. Therefore, the proposal includes a flexible topology, using both radio as fiber in the backhaul and access to sensor nodes via a wireless sensor network. For demonstration of the proposed, tests were done on a channel emulation bench using fiber backhaul between the base and a repeater element. These tests were performed on the 915 MHz band, the FSK varying its transmission rate to assess the coverage area for each rate. As a result curves were obtained showing that for a fixed RSSI, the higher the worse transmission rate is the bit error rate. And the higher the lowest environmental attenuation factor will be the coverage area at all rates tested. Therefore the lower the reception power, the greater the distance between the sensors. Also tests were made in a real environment to evaluate the protocols implemented. It consists in the access of the node sensors of the clusters through multiple hops and reception power measurements of each sensor node and a packet loss rate. Through testing it has verified the operation of the proposal and implemented the flexible topology proposal.

Keywords: Wireless sensor network. Optical fiber. Multihop. Hybrid topology

LISTA DE FIGURAS

Figura 1: Topologia de solução geral.	19
Figura 2: Arquitetura de múltiplos saltos de uma rede de sensores sem fio (HOSSSEN <i>et al.</i> , 2009).	20
Figura 3: <i>Backhaul</i> de rádio sobre fibra. (HOSSSEN <i>et al.</i> , 2009)	21
Figura 4: Sistema de acesso aos nós sensores. (ASSUMPÇÃO, 2011).....	22
Figura 5: Topologia do sistema implementado. (ANDRADE, 2013).....	23
Figura 6 - Diagrama do sistema de comunicação.....	25
Figura 7: Diagrama do modelo <i>Log - distance</i>	28
Figura 8- Canal AWGN	29
Figura 9: Curva característica da modulação FSK	32
Figura 10: Diagrama em blocos.	33
Figura 11: Diagrama do sistema de RoF.....	34
Figura 12: Topologia geral da solução proposta.....	36
Figura 13: Resultados obtidos para a modulação FSK.(ASSUMPÇÃO, 2011)	38
Figura 14: Visão funcional da topologia proposta.	39
Figura 15: Camadas implementadas na topologia.....	40
Figura 16: Visão temporal.	42
Figura 17: Diagrama em blocos da bancada.	44
Figura 18: Bancada de transmissão.....	45
Figura 19: Bancada de recepção.	45
Figura 20: Atenuador utilizado RVA -3000.	46
Figura 21: Esquemático equivalente do RVA -3000 (Mini-Circuits).	46
Figura 22: Tensão X Atenuação em 915 MHz.....	47
Figura 23: Curva característica medida do atenuador.....	47
Figura 24: Chave de RF Modelo ZMSW – 1111	48
Figura 25: Bancada de teste com chave de RF.....	49
Figura 26: Nó sensor e base.	50
Figura 27: Repetidor utilizado.....	51
Figura 28: Setup para verificar a potência dos rádios.....	52
Figura 29: Payload fixo de 52 bytes.	53
Figura 30: Pacote utilizado.....	53
Figura 31: Fluxograma da base.....	55
Figura 32: Fluxograma nó sensor.....	56
Figura 33: Fluxograma repetidor nó B.	57
Figura 34: Fluxograma repetidor nó C.....	58

Figura 35: Software Smart RF Studio.....	59
Figura 36: Função dos registradores.....	60
Figura 37: Fluxograma do software de coleta da bancada.	62
Figura 38: Topologia do teste externo.	64
Figura 39: Visão do teste externo.....	65
Figura 40: Visão dos RCH externos.	66
Figura 41: Nós sensores do cluster 1.....	66
Figura 42: RCH2.	67
Figura 43: Antena painel de 90°.	67
Figura 44: Antena Omni vertical.	68
Figura 45: Pacote utilizado.....	69
Figura 46: Fluxograma do software de coleta do teste externo.	71
Figura 47: Topologia com <i>backhaul</i> de fibra óptica.	73
Figura 48: Gráfico Prx X BER do resultado dos testes em bancada para a modulação FSK.....	77
Figura 49: Distâncias alcançadas em relação as BER's obtidas para as taxas testadas.	79

LISTA DE TABELAS

Tabela 1: Resultados para a modulação FSK (ASSUMPÇÃO, 2011).....	22
Tabela 2: Características do fator β . (<i>RAPPAPORT, 2002</i>).....	29
Tabela 3: Parâmetros para a modulação FSK. (Ieee, 2006).....	31
Tabela 4: Resultados obtidos.....	75
Tabela 5: Resultados dos nós sensores do cluster 1.	80
Tabela 6: Resultados dos nós sensores do cluster 2.	81

LISTA DE ABREVIATURAS

AVGN = *Additive white Gaussian noise*

BER = *Bit Error Rate*

EIN = *Equivalent Input Noise*

FSK = *Frequency-Shift Keying*

LNA = *Low Noise Amplifier*

PER = *Packet Error Rate*

RCH = *Repetidor Cluster Head*

RoF = *Radio Over Fiber*

RF = *Rádio Frequência*

RSSF = *Rede de sensores sem fio*

RSSI = *Received Signal Strength Indicator*

SUMÁRIO

1	INTRODUÇÃO.....	15
1.1.	Objetivo.....	16
1.2.	Motivação.....	17
1.3.	Organização.....	17
2	TRABALHOS RELACIONADOS.....	19
3	TECNOLOGIA RÁDIO E FIBRA ÓPTICA.....	25
3.1	Sistema de comunicação.....	25
3.1.1	Modelo de Propagação.....	26
3.1.1.1	Espaço livre.....	27
3.1.1.2	Log - Distance.....	27
3.1.2	Potência de ruído.....	29
3.1.3	Cálculo da Sensibilidade.....	30
3.1.4	Modulação.....	31
3.2	Redes de sensores sem fio.....	33
3.3	Sistema de rádio sobre fibra.....	34
4	SOLUÇÃO PROPOSTA.....	36
4.1	Análise da cobertura.....	37
4.2	Acesso aos nós sensores.....	39
5	MATERIAIS E MÉTODOS.....	43
5.1	Testes em bancada.....	43
5.1.1	Bancada de testes.....	43
5.1.2	Emulação do canal.....	45
5.1.4	Plataforma de sensores.....	49
5.1.4.1	Repetidor utilizado.....	51
5.1.4.2	Aferição dos rádios.....	51
5.1.5	Pacote utilizado.....	52
5.1.6	Firmware.....	54
5.1.7	Configuração dos registradores.....	58
5.1.8	Software de coleta.....	61
5.1.9	Experimento realizado na bancada.....	62
5.2	Testes externos.....	63
5.2.1	Cenário montado.....	64
	64
5.2.2	Repetidor <i>cluster head</i>	67
5.2.3	Antenas utilizadas.....	67
5.2.4	Pacote utilizado.....	68
5.2.5	Firmware.....	69
5.2.6	Software de coleta dos testes externos.....	70
5.2.7	Experimento realizado.....	72

6 ANÁLISE DE CENÁRIOS DE APLICAÇÃO.....	73
6.1 Exemplo Aplicado	74
7 RESULTADOS.....	77
7.1 Resultados dos testes da bancada.....	77
7.2 Resultados dos testes no ambiente externo	79
8 CONCLUSÃO	83
8.1. Contribuições	84
8.2. Propostas de trabalhos futuros	84
9 REFERÊNCIAS	85

1 INTRODUÇÃO

As redes de sensores sem fio estão se tornando uma realidade, sendo que estes sensores são espalhados por uma certa região e devem ser aglutinados de alguma forma, para que as grandezas monitoradas e os controles necessários sejam encaminhados de forma eficiente a um centro de gerência.

Em alguns trabalhos os sensores podem ser aglutinados através de sub-redes de acordo com a sua energia, ou seja, eles podem se comunicar com os nós sensores centrais durante um período, mas podem se comunicar com outros em outras circunstâncias, por exemplo, quando existe algum problema com algum nó central. (DING e YAMAUCHI, 2010)

A comunicação entre os nós sensores pode ser realizada através de múltiplos saltos, sendo que nesta vertente existem trabalhos que avaliam uma rede de sensores e o consumo de energia de cada nó sensor, sendo que a distância mínima entre os nós sensores interfere na eficiência de energia nos múltiplos saltos. (FEDERO e COLLIER, 2007)

Estes sensores dispersos podem ser aglutinados em clusters que se unem em células, que podem ser interligados a uma central de gerência. A interligação destas células com a base pode ser feita via rádio, mais comum de ser utilizado, ou via fibra usando a tecnologia de rádio sobre fibra.

Em (ASSUMPÇÃO, 2011) foi utilizada a estratégia de um sistema de rádio sobre fibra (RoF), em que os nós sensores das células se comunicavam via rádio com o RoF, em que foi avaliado a área de cobertura das células e o impacto deste sistema utilizando um ou dois RoF em série, com e sem amplificador de baixo ruído (LNA). Para este mesmo sistema, em (ANDRADE, 2013) foram implementados dois protocolos SPP-MAC (*Scheduled Priority Polling Medium Access Control*) e HMARS (*Hybrid Medium Access Control for Hybrid Radio-over-Fiber Wireless Sensor Network Architecture*), que levam em consideração o

atraso ocasionado pela fibra, colisões entre os pacotes e a eficiência de energia das redes.

Este trabalho propõe uma estratégia de acesso aos nós sensores considerando vários *clusters* que se aglutinam formando uma célula que se comunica a um centro de gerência. A topologia para interligar as células formadas por vários clusters, denominada como *backhaul*, poderá ser com fibra óptica ou rádio, enquanto a comunicação nas células entre os nós sensores será via rádio através de múltiplos saltos. Portanto esta topologia proposta é uma solução híbrida.

Uma estratégia semelhante foi adotada em (HOSSEN *et al.*, 2009) ao propor um sistema com *backhaul* de fibra óptica para a comunicação dos *clusters head* e com múltiplos saltos entre os nós sensores até o *cluster head*. Entretanto a proposta não apresenta uma solução real implementada e com possibilidade de ser híbrida.

Na proposta aqui apresentada, foram feitos testes em um ambiente real implementando esta topologia de acesso aos nós sensores através de múltiplos saltos, utilizando um repetidor *cluster head* (RCH). Este RCH tem como funções: ser um repetidor no canal de comunicação entre os RCH dos clusters da célula e ser um *cluster head*, concentrando as informações dos nós sensores do cluster.

Foram realizados testes também em uma bancada montada em laboratório para emular o canal sem fio utilizando backhaul de fibra óptica, com o elemento repetidor e variando as taxas utilizadas. Desta forma, foi possível dimensionar a área de cobertura das redes de sensores sem fio (RSSF) que são utilizadas na topologia da solução proposta neste estudo.

1.1. Objetivo

O objetivo deste trabalho é propor uma topologia híbrida de acesso a nós sensores que estão espalhados em uma certa área de forma a concentrar

estas informações em um centro de gerência. Esta topologia utiliza comunicação rádio nas redes de sensores sem fio e entre os repetidores com os RCH, enquanto no *backhaul* entre os repetidores e a base pode ser utilizada fibra óptica ou rádio.

1.2. Motivação

Criar estratégias com novas topologias para interligação de conjuntos de sensores, denominados clusters, com um centro de gerências.

1.3. Organização

Esta dissertação está organizada, como se segue:

No capítulo 2 são apresentados os trabalhos relacionados com a topologia de solução geral apresentada e sendo estes motivadores do que foi proposto.

No capítulo 3 são apresentados os sistemas de rádio e fibra óptica detalhados com suas fórmulas que serão utilizadas ao longo do trabalho.

No capítulo 4 é apresentada a solução proposta em três visões: lógica, em camadas e temporal. Sendo feita a explicação de forma exemplificada para acessar um nó sensor de um cluster.

No capítulo 5, os materiais e métodos são apresentados, descrevendo a metodologia aplicada nos testes na bancada de emulação montada em laboratório e os testes externos realizados para validação da forma de acesso aos nós sensores por múltiplos saltos. Sendo também detalhados os equipamentos utilizados, firmwares e softwares de coleta de dados.

No capítulo 6 é demonstrado um cenário para o monitoramento de equipamentos de energia utilizando a topologia proposta e mensurando o número de células possíveis.

No capítulo 7 são apresentados os resultados obtidos na bancada de emulação sendo correlacionada a potência de recepção em relação à taxa de erros de bit para cada taxa de transmissão utilizada e a distância da área de cobertura. Também foram obtidas tabelas com a potência de recepção e taxa de perda de pacotes nos testes realizados no ambiente externo realizando o acesso aos nós sensores por múltiplos saltos.

No capítulo 8 é apresentada a conclusão referente aos resultados obtidos neste trabalho, assim como as contribuições e propostas de trabalhos futuros.

O trabalho se encerra com as respectivas referências e anexos.

2 TRABALHOS RELACIONADOS

Com o crescimento do monitoramento de grandezas em ambientes, tais como: temperatura, umidade, pressão, CO₂, através de sensores no meio urbano, em indústrias, edificações, gasodutos (MELLO *et al.*, 2007); trazendo-se a necessidade da aquisição e armazenamento destes dados para um centro de gerência. Desta forma torna-se possível realizar a análise e tomada de decisão através destas informações.

Para resolver esta necessidade de enviar as informações coletadas por nós sensores dispersos em uma área existem algumas estratégias que podem ser adotadas, como no caso da topologia da Figura 1, em que os nós sensores estão aglutinados em clusters enviando seus pacotes para um concentrador. Os concentradores RCH se comunicam entre si e com os repetidores, os quais através do *backhaul* irão enviar as informações para a base. Esta base irá enviar os dados respondidos pelos nós sensores para o centro de gerência.

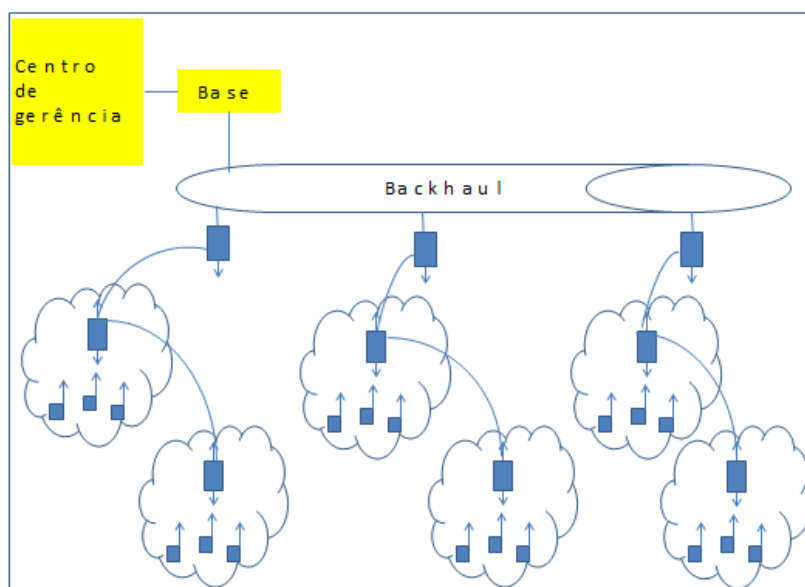


Figura 1: Topologia de solução geral.

Na estratégia proposta na topologia da Figura 1, a conectividade pode ser dividida em duas partes: acesso aos nós sensores e *backhaul* de interligação.

Para realizar o acesso ao nó sensor pelo *RCH* e entre os mesmos, a comunicação é via rádio, sendo avaliada a distância de cobertura possível e a estratégia para os clusters se comunicarem através de saltos. Enquanto para a *backhaul* de rádio ou fibra óptica, será avaliado o número de células possíveis ao longo do enlace.

No cenário em que é necessário aglutinar os nós sensores dispersos e enviar suas informações a um centro de gerência, foi proposta a estratégia em (HOSSEN *et al.*, 2009) de enviar as informações dos nós sensores através de múltiplos saltos para um concentrador (PNC), como pode ser visto na Figura 2.

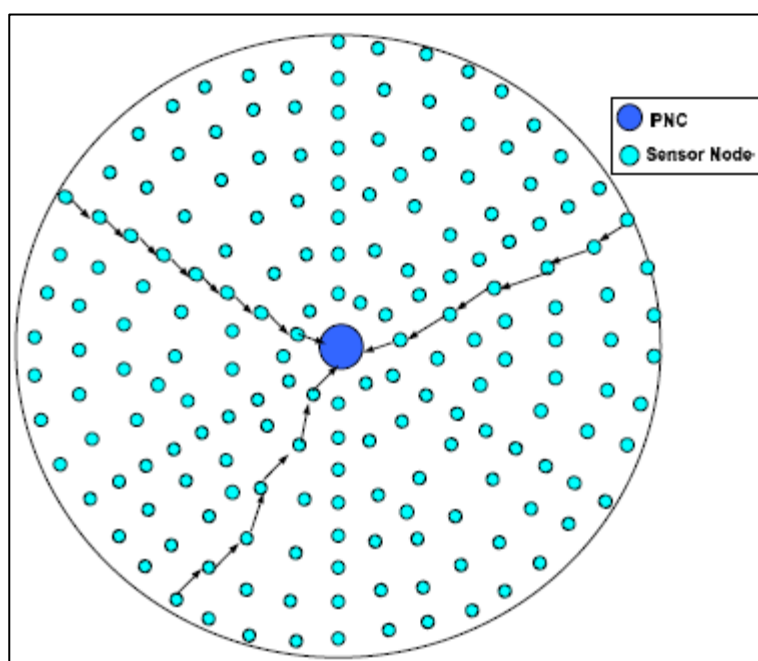


Figura 2: Arquitetura de múltiplos saltos de uma rede de sensores sem fio (HOSSEN *et al.*, 2009).

Este concentrador PNC deverá se comunicar com outros PNC de outras células, sendo que esta comunicação será feita através de um *backhaul* de fibra óptica como pode ser visto na Figura 3.

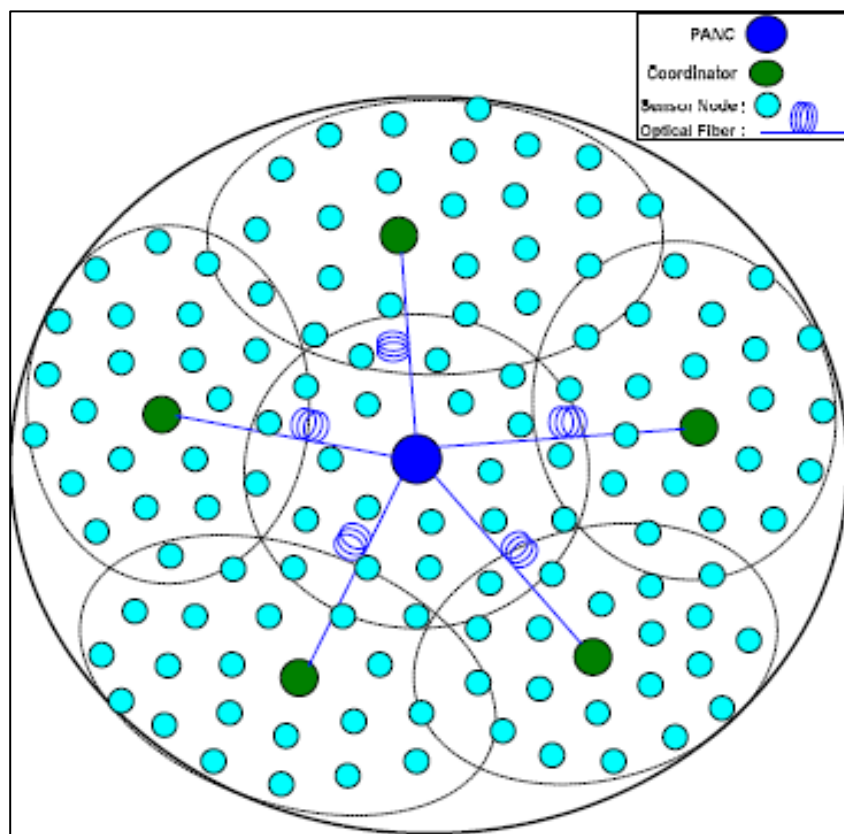


Figura 3: *Backhaul* de rádio sobre fibra. (HOSSEN *et al.*, 2009)

Na estratégia proposta em (HOSSEN *et al.*, 2009) está sendo utilizada a tecnologia de rádio sobre fibra para ampliar a área de cobertura das redes de sensores sem fio. Na análise do impacto em uma rede de sensores sem fio, utilizando a tecnologia de rádio sobre fibra, foi desenvolvido o trabalho de (ASSUMPÇÃO, 2011). Neste trabalho, o sistema proposto utiliza uma central de controle que requisita os dados através da base que passa por um *backhaul* de fibra óptica até chegar às redes de sensores em fio, como pode ser visto na Figura 4.

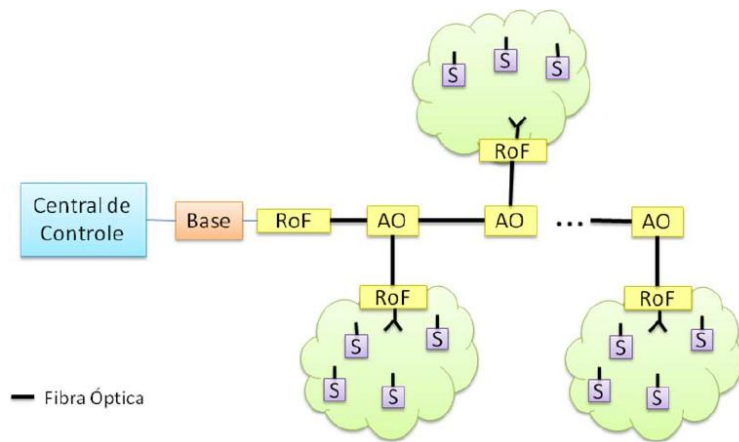


Figura 4: Sistema de acesso aos nós sensores. (ASSUMPÇÃO, 2011)

Considerando o sistema da Figura 4, foram realizados testes em uma bancada de emulação de canal. Foram montados alguns cenários para realizar os experimentos: Considerando um *backhaul* sem RoF, com 1 ou 2 RoF e utilizando ou não amplificador de baixo ruído (LNA).

Foi possível avaliar através dos cenários de testes as distâncias de cobertura das redes de sensores e potência de recepção considerando as modulações MSK e FSK, sendo que para esta última os resultados obtidos estão na Tabela 1.

Tabela 1: Resultados para a modulação FSK (ASSUMPÇÃO, 2011)

FSK	P_{rx} experimental (dBm)	P_{rx} a partir dos cálculos (dBm)	Distância de Cobertura experimental (m)	Distância de cobertura a partir dos cálculos (m)
Sem RoF	-95	-	141,4	-
1 RoF	-71	-68,36	10,2	8,5
2 RoF	-68,66	-66,6	8,7	7,5
Sem RoF Com LNA	-90,5	-	259,6	-
1 RoF Com LNA	-71,9	-68,36	26,9	21,1
2 RoF Com LNA	-67,3	-66,6	19,7	18,8

Nesta proposta, houve o fator limitante da baixa cobertura na célula. O motivo disso deve-se à antena estar ligada diretamente ao RoF e ao fato de ocorrer uma perda devido à figura de ruído ao longo do *backhaul*.

Através deste trabalho foi analisado o impacto na distância na área de cobertura das redes de sensores sem fio devido ao RoF e comprovou-se a possibilidade de implementar a tecnologia de rádio sobre fibra.

No trabalho (ANDRADE, 2013) foi utilizada a mesma topologia do trabalho de (ASSUMPÇÃO, 2011), como pode ser visto na Figura 5. Houve, neste caso, a preocupação do atraso adicional de propagação dos sinais, devido à grande distância dos enlaces de fibra óptica, uma vez que o atraso pode exceder o limite de temporização dos protocolos de acesso ao meio. Este problema pode ser mitigado com utilização de protocolos MAC centralizado com ajustes de propagação entre os clusters, para evitar colisões.

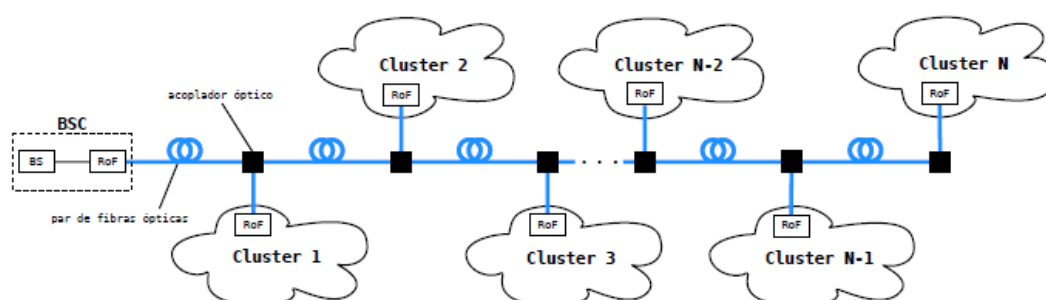


Figura 5: Topologia do sistema implementado. (ANDRADE, 2013).

Neste trabalho (ANDRADE, 2013) foram desenvolvidos dois protocolos de acesso ao meio:

- SPP-MAC (*Scheduling of Pooling Priority Medium Access Control*), protocolo centralizado, ou seja, a base tem o conhecimento de toda a topologia, sendo que utiliza a sinalização para o envio dos dados dos nós sensores, e pode utilizar de prioridades para os nós sensores da rede. Preocupa-se também com o desperdício de energia.
- H-MARS (*Híbrid Medium Access Control for Hybrid Radio-over-Fiber Wireless Sensor Network*), foi desenvolvido para redes híbridas, de redes de sensores sobre fibra óptica para melhorar a economia de energia utilizada. Sistema de acesso híbrido por

Timing Division Multiplexing (TDMA) e Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) não persistente.

Foram feitas simulações utilizando estes protocolos e implementado em laboratório utilizando a topologia proposta na Figura 5.

3 TECNOLOGIA RÁDIO E FIBRA ÓPTICA

Neste capítulo são apresentados os conceitos necessários dos sistemas de rádio e fibra óptica, para apresentar a proposta deste trabalho.

3.1 Sistema de comunicação

O sistema de comunicação das redes de sensores sem fio é formado por um elemento transmissor e outro receptor que estão interligados a antenas como pode ser visto na Figura 6. Eles se comunicam através de um link rádio que deve ser analisado para avaliar a sua área de cobertura das células através da potência de sinal do rádio, antenas adequadas e modulação. Estes parâmetros citados assim como os da Figura 6 que estão descritos abaixo são utilizados para verificar o funcionamento deste link de forma adequada. Este cálculo é conhecido como *Link Budget*.

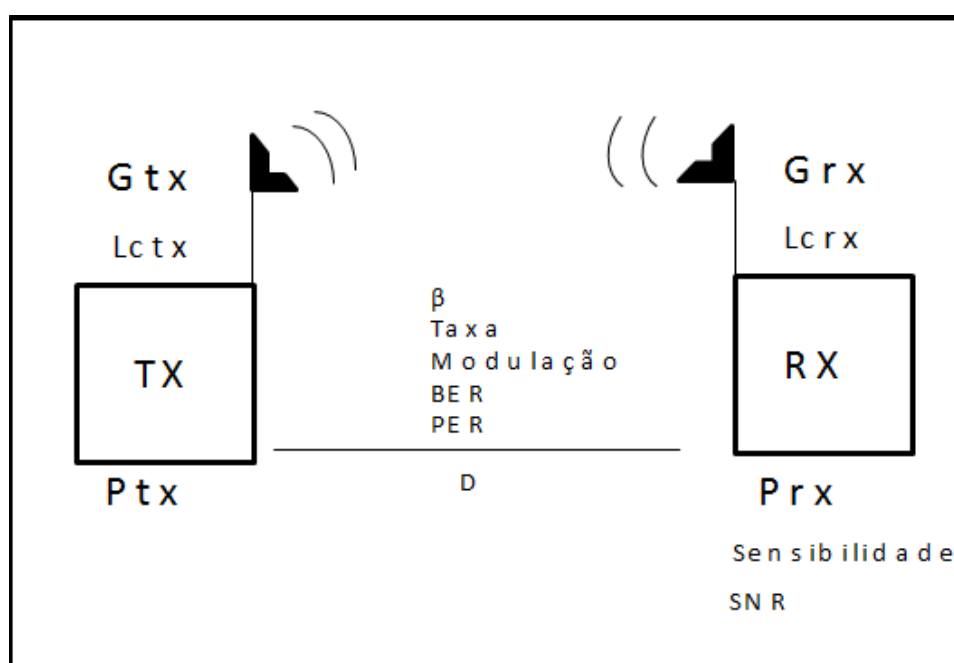


Figura 6 - Diagrama do sistema de comunicação

- Ptx: Potência de transmissão.
- Gtx: Ganho da antena de transmissão.
- Lctx: Atenuação dos cabos na transmissão.
- D: Distância entre o transmissor e o receptor.
- β : Índice que caracteriza o ambiente.
- R: Taxa de transmissão entre os rádios.
- M: Modulação utilizada.
- BER: Taxa de erro de bit.
- PER: Taxa de erro de pacote.
- Prx: Potência de recepção.
- Grx: Ganho da antena na recepção.
- Lcrx: Atenuação dos cabos na recepção.
- S: Sensibilidade, ou seja, a menor potência de recepção para obter uma BER específica que é encontrada em função de uma modulação.
- SNR: Relação entre a potência do sinal e o ruído.

Através deste diagrama pode-se verificar que para garantir uma comunicação de rádio com a maior BER possível, devem ser determinados os ganhos das antenas e potência de transmissão, para conseguir uma sensibilidade mínima, considerando a distância entre as antenas, frequência e ambiente.

3.1.1 Modelo de Propagação

Os modelos de propagação serão utilizados para realizar o cálculo da potência de sinal recebido no receptor através dos parâmetros informados no sistema de comunicação acima. Através destes modelos é possível encontrar a potência de recepção de um sistema de acordo com os parâmetros descritos no diagrama de *link budget* acima.

Estes modelos serão utilizados para calcular a distância de cobertura entre os nós sensores.

3.1.1.1 Espaço livre

O modelo de propagação no espaço livre considera que o sistema de comunicação possui linha de visada e o ambiente está totalmente desobstruído. Este modelo é conhecido também como Friss, e sua Equação 1 (RAPPAPORT, 2002) possibilita realizar o cálculo da potência de sinal recebida, atenuação do espaço livre e a área de cobertura.

$$Prx = Ptx + Gtx + Grx - Lel \quad (1)$$

Onde Prx a potência de recepção, Ptx potência de transmissão, Gtx ganho da antena na transmissão, Grx ganho da antena na recepção e Lel a atenuação no espaço livre. Esta atenuação pode ser calculada pela Equação 2.

$$Lel = 10 \log \left(\frac{4\pi d}{\lambda} \right)^2 \quad (2)$$

$$Prx = Ptx + Gtx + Grx - 10 \log \left(\frac{4\pi d}{\lambda} \right)^2 \quad (3)$$

Onde a distância é indicada pela variável d, λ o comprimento de onda da frequência utilizada. As variáveis Ptx e Prx são em dBm e as variáveis Gtx e Grx são em dBi.

3.1.1.2 Log - Distance

O modelo *Log-distance* (RAPPAPORT, 2002) da Figura 7 é utilizado para considerar o tipo de ambiente. Este modelo leva em consideração a potência do sinal para uma distância d_0 de referência, que deve ser ao menos 10 vezes menor que a distância d (RAPPAPORT, 2002). A partir da potência de recepção para esta distância de referência e do fator β que caracteriza a atenuação do ambiente, é possível calcular a potência de recepção na distância desejada.



Figura 7: Diagrama do modelo Log - distance.

A Equação 4 calcula a potência de recepção para a distância pela soma da potência de transmissão P_{tx} , ganho das antenas G_{tx} e G_{rx} , e atenuação total.

$$Prx(d) = P_{tx} + G_{tx} + G_{rx} + L_{tot} \quad (4)$$

Para este modelo a atenuação total L_{tot} , ou *path loss*, pode ser calculada pela Equação 5, que considera a distância total d , o comprimento de onda da frequência utilizada λ , o fator de atenuação do ambiente β e a distância de referência d_0 .

$$L_{tot} = 10 \log \left(\frac{4\pi d}{\lambda} \right)^2 + 10\beta \log \left(\frac{d}{d_0} \right) \quad (5)$$

$$Prx(d) = P_{tx} + G_{tx} + G_{rx} - 10 \log \left(\frac{4\pi d}{\lambda} \right)^2 - 10\beta \log \left(\frac{d}{d_0} \right) \quad (6)$$

$$Prx(d) = Prx(d_0) - 10\beta \log \left(\frac{d}{d_0} \right) \quad (7)$$

A partir da Equação 7 é possível isolar a variável d , para determinar a distância de cobertura da RSSF, da base até o sensor, como na Equação 8.

$$d = d_0 * 10^{\frac{P_{tx} + G_{tx} + G_{rx} - 10 \log \left(\frac{4\pi d_0}{\lambda} \right) - Prx}{10 * \beta}} \quad (8)$$

Na Equação 8 em que se calcula a distância, temos como variável o β que representa um fator de caracterização dos ambientes, como pode ser visto na Tabela 2. Sendo que quanto maior o seu valor maior é a atenuação do ambiente.

Tabela 2: Características do fator β . (RAPPAPORT, 2002)

β	Ambiente
1,6 á 1,8	Dentro de construção com linha de visada.
4 á 6	Dentro de construção sem linha de visada.
3,41	Área aberta

3.1.2 Potência de ruído

A comunicação rádio quando encontra-se em um ambiente controlado, como em uma bancada de emulação de canal. Nestes casos é necessário considerar a potência do ruído, ou seja, um canal *Additive White Gaussian Noise* (AWGN) que tem adição de ruído, conforme a Figura 8.

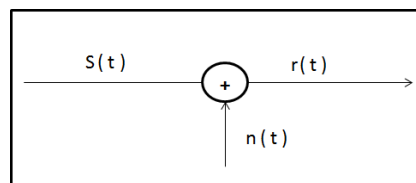


Figura 8- Canal AWGN

Nesta figura temos o sinal $s(t)$ que tendo a adição da potência do ruído $n(t)$, gera como resultado o sinal $r(t)$, que possui uma relação sinal-ruído.

Este ruído que foi adicionado, refere-se a um ruído de agitação térmica que pode ser calculado pela Equação 9.

$$N = K * T * B \quad (9)$$

Onde K é a constante de Boltzman $1,38 \times 10^{-23}$ J/K, T temperatura em Kelvin e B é a banda ocupada em Hz.

A potência de ruído total, na Equação 9, pode ser calculada pela somatória do ruído térmico da Equação 10, com a figura de ruído do receptor.

$$N_t[dBm] = N[dBm] + NF[dB] \quad (10)$$

3.1.3 Cálculo da Sensibilidade

Este trabalho será relacionado à sensibilidade do rádio, ou seja, a potência do sinal recebida para uma determinada BER, e da distância com a taxa de erros de bit (BER). Desta forma é necessário calcular a relação sinal-ruído (SNR), pois para toda SNR temos uma energia do bit por N0, E_b/N_0 , relacionado com a BER para uma modulação. (KARL e WILLIG, 2005)

A sensibilidade do rádio é determinada pela Equação 11, considerando uma BER específica.

$$S = E_b * R \quad (11)$$

Onde E_b é a energia do bit, e pode ser calculada pela integral do bit. A taxa R como mostrada na Equação 12, é o inverso do período de duração do bit T_b .

$$R = \frac{1}{T_b} \quad (12)$$

A densidade espectral da potência de ruído N_0 pode ser calculada pela Equação 13, onde N é a potência do ruído e Bw a banda ocupada para uma determinada modulação.

$$N_0 = \frac{N}{B_w} \quad (13)$$

A potência de ruído N é a somatória do ruído térmico da Equação 9, com a figura de ruído NF informada no datasheet pelo fabricante do receptor.

Relacionando estas fórmulas chegamos a SNR na Equação 14 e à sensibilidade calculada para uma determinada BER na Equação 15.

$$\frac{S}{N} = \frac{Eb}{N0} * \frac{R}{Bw} \quad (14)$$

$$S = \frac{Eb}{N0} * R \left(K * T + \frac{NF}{Bw} \right) \quad (15)$$

Esta relação da sensibilidade com a BER depende da modulação utilizada e da taxa utilizada.

3.1.4 Modulação

A Modulação *Frequency Shift Key* (FSK), é utilizada para redes sem fio (Akyildiz *et al.*, 2002) assim como está incluída no padrão 802.15.4g (Ieee, 2006). Por este motivo foi à modulação utilizada nas redes de sensores montadas neste trabalho.

Esta é a única modulação em que existe uma flexibilidade entre a banda ocupada e a taxa, permitindo ocupações maiores de banda com menores taxas, atendendo a resolução 506 (ANATEL, 2008). A Tabela 3 apresenta como referência algumas modulações utilizadas pelo padrão 802.15.4g (Ieee, 2006) O presente trabalho utiliza a modulação FSK e, portanto, está aderente com o padrão 802.15.4.

Tabela 3: Parâmetros para a modulação FSK. (Ieee, 2006)

PHY MHz)	Banda de Frequência (MHz)	Parâmetros distribuídos		Parâmetros de dados		
		Taxa Chip kchips/s)	Modulação	Taxa Bit (Kb/s)	Taxa Symbol (Ksymbol/s)	Symbols
863	863-870			50	50	Binário
			Filtrado 2FSK	100	100	
			Filtrado 4FSK	200	200	4-ary
			OFDM	Como definido em 18.2		
868	868-870	100	O-QPSK	625-50 Como definido em 18.3	3.125	
896	896-901		Filtrado 2FSK	10	10	Binário

				20	20	
				40	40	
901	901-902		Filtrado 2FSK	10	10	Binário
				20	20	
				40	40	
915	902-928		Filtrado 2FSK	50	50	Binário
			Filtrado 2FSK	150	150	
			Filtrado 2FSK	200	200	

A Equação 16 relaciona a Probabilidade de erro (P_e), ou taxa de erro de bits (BER), com o E_b/N_0 para a modulação FSK (LATHI, 1998).

$$P_{e,FSK} = \frac{1}{2} e^{\left(-\frac{E_b}{N_0}\right)} \quad (16)$$

Considerando a Equação 16, obtêm-se a curva característica da probabilidade de erro (P_e), ou taxa de erros de bits (BER) em relação ao E_b/N_0 para a modulação FSK, como pode ser visto na Figura 9.

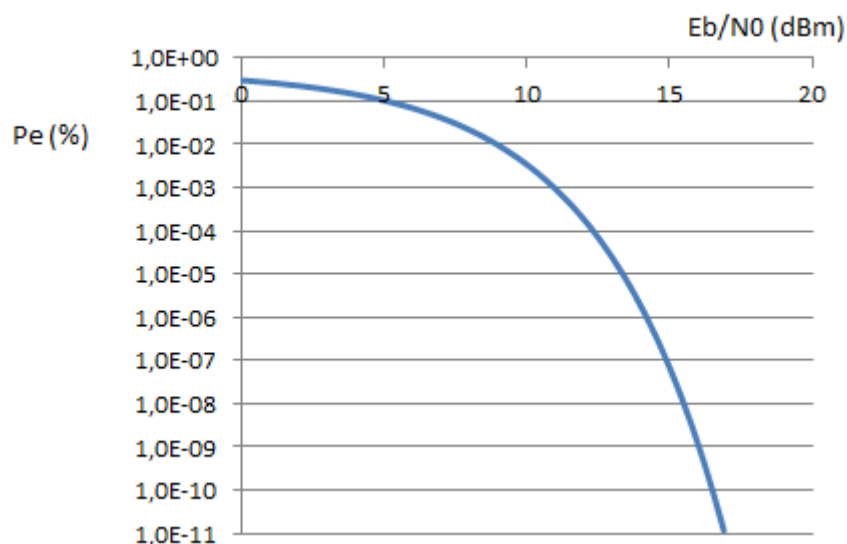


Figura 9: Curva característica da modulação FSK

A taxa de erros por pacotes (PER) relaciona com a BER, e com o número de bits enviados do pacote n , através da equação 17 abaixo:

$$PER = 1 - (1 - BER)^n \quad (17)$$

3.2 Redes de sensores sem fio

As redes de sensores sem fio são formadas por equipamentos capazes de monitorar grandezas, tais como: gasodutos, oleodutos, cidades e transmitir estas informações através de sensores sem fio para onde podem ser analisadas.

Os nós sensores utilizados neste trabalho são formados por 4 componentes principais (RADIUINO, 2014), como pode ser visto no diagrama da Figura 10.

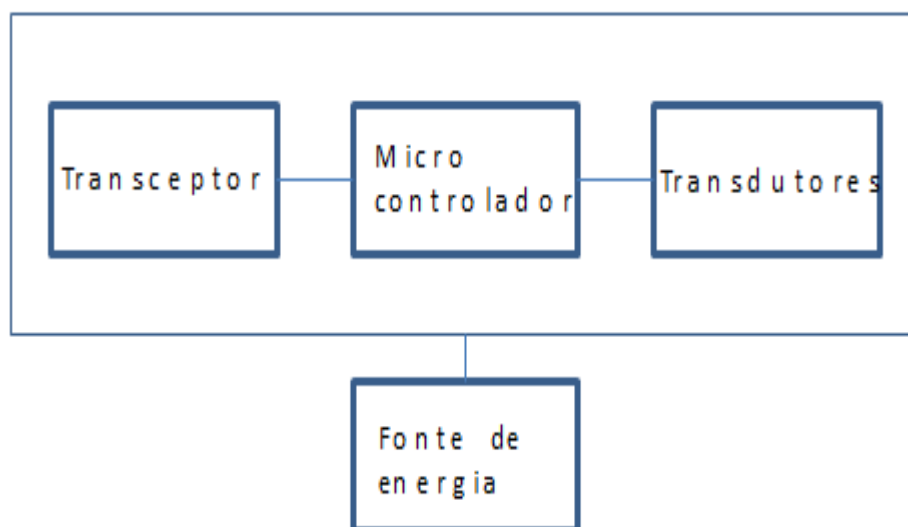


Figura 10: Diagrama em blocos.

- Transceptor: Responsável pela comunicação sem fio.
- Micro controlador: Local em que ocorre o armazenamento e processamento das informações do sensor.
- Transdutores: Dispositivos que realizam a interação com o meio e fazem a captação das grandezas que vão ser trabalhadas pelo micro controlador.
- Fonte de alimentação: Responsável por fornecer energia ao equipamento.

3.3 Sistema de rádio sobre fibra

No sistema de RoF o sinal de radiofrequência modula um laser, podendo ser direto ou externo esta modulação (SILVA, 2009). A Figura 11 apresenta os elementos básicos de um sistema RoF.

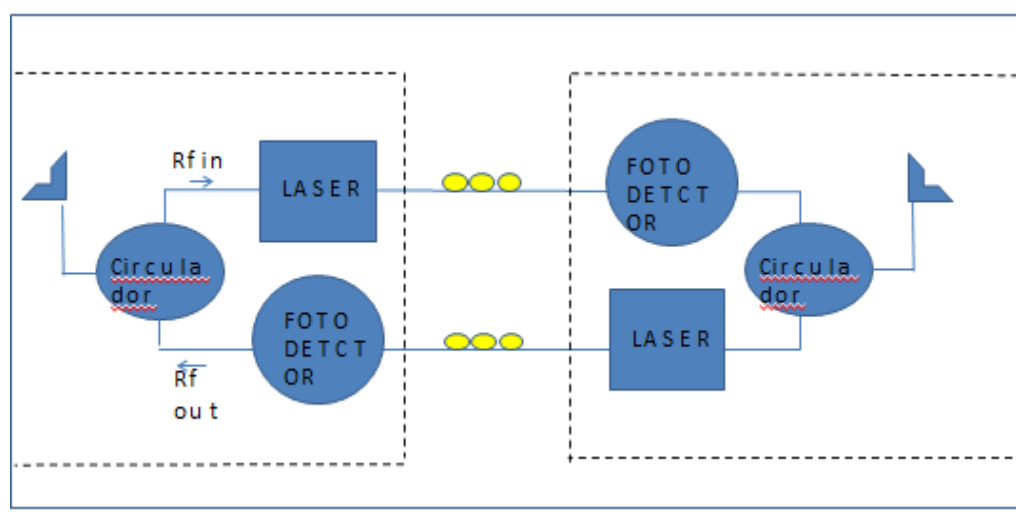


Figura 11: Diagrama do sistema de RoF.

O sinal de radio frequência (RF) de entrada modula o laser que é transmitido pela fibra. Após a transmissão, o sinal percorre a fibra óptica e no final do enlace existe um foto detector que converte o sinal óptico ao sinal de RF original. Para que seja utilizada a mesma antena para transmissão e recepção, é necessário o circulador, que tem como função evitar a realimentação do RoF com o sinal de rádio base. O sinal ao sair do circulador modula o sinal do laser para retornar ao enlace de fibra óptica.

Neste sistema de rádio sobre fibra existe a interferência do ruído causado pelo RoF. Esta figura é representada pelo ruído equivalente de entrada (EIN), que é representado pela Equação 18 (ASSUMPÇÃO, 2011).

$$EIN = Rrin + Rshot + Rt \quad (18)$$

Sendo que:

- R_{rin} : Ruído de intensidade relativa do laser, sendo detectado no receptor.
- R_{shot} : Ruído de origem da corrente do fotorreceptor.
- R_t : Ruído térmico proveniente da temperatura dos equipamentos.

A potência do ruído é aumentada a cada novo equipamento de RoF que é inserido no *backhaul*, o que degrada a relação sinal-ruído e não pode ser melhorada através de amplificadores, pois ao se ampliar a potência do sinal estará sendo ampliada a potência do ruído também (LATHI, 1998).

Para realizar a análise do número de células possível ao longo do *backhaul* de fibra é necessário saber a potência de recepção, a inserção da figura de ruído a cada RoF e a divisão da potência do sinal a cada acoplador óptico. Através da Equação 19 é possível determinar a potência de recepção, ou seja a sensibilidade da base, para viabilidade da rede, considerando n como o número de células de sensores.

$$Prx = [\sum_{i=1}^n EIN + 10 \log(\text{Bhz})] + SNR_{dB} \quad (19)$$

A potência de recepção do sistema leva em consideração a somatória da figura de ruído de todos os RoF's, atenuação da fibra e atenuação devido a divisão da potência do sinal em 50% a cada acoplador óptico. O sistema deve desconsiderar o acoplador para a última célula assim como a perda para os conectores ópticos, e a distância da fibra está representado por d , demonstrado na Equação 20, em que considera apenas para a última célula que representa o pior caso.

$$Prx_{up} = Ptx - (d * 0,4 \text{ dB}) - ((n - 1) * 6\text{dB}) \quad (20)$$

Na Equação 20 está sendo considerado a atenuação de 0,4 dB por quilômetro de fibra e a atenuação de 6 dB por acoplador óptico ao longo do *backhaul*.

4 SOLUÇÃO PROPOSTA

Como observado nos capítulos anteriores, existem propostas de solução para o envio das informações para o centro de gerência através de RSSF. Porém, estas propostas não são flexíveis para a criação de estratégia de interligação de clusters de sensores espalhados no ambiente, tendo também problema na área de cobertura das redes de sensores sem fio.

Considerando que existem várias informações sendo coletadas por nós sensores dispersos e a necessidade de centralizá-las em um centro de gerência para monitoramento e análise, foi proposta uma topologia híbrida para acesso aos nós sensores dispersos, como na Figura 12.

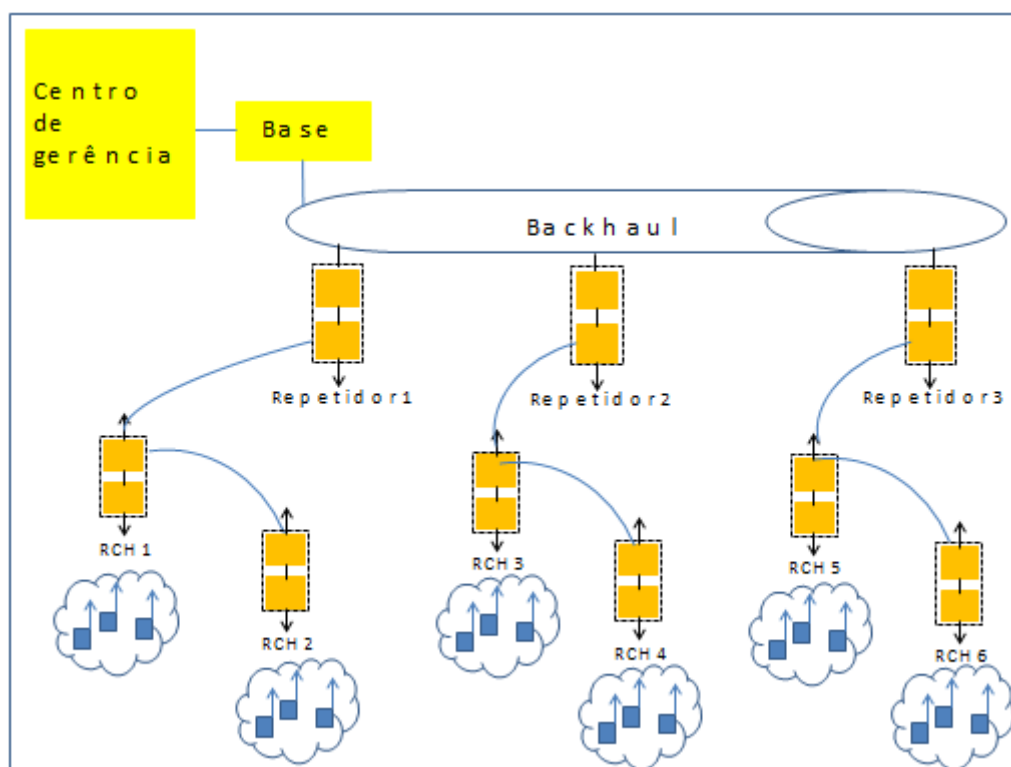


Figura 12: Topologia geral da solução proposta.

Nesta topologia, os nós sensores dispersos foram aglutinados em clusters. As informações destes clusters são acessadas pelo repetidor/cluster

head (RCH), que possui dois rádios, sendo o primeiro para repetir os pacotes quando necessários para o RCH do próximo cluster e o segundo para concentrar de forma a transmitir e receber os dados para os nós sensores do cluster.

A forma de acesso dos nós sensores aos RCH, e dos mesmos com os repetidores, será sempre por comunicação sem fio. Enquanto o *backhaul* de comunicação entre os repetidores com a base será via rádio ou fibra óptica, ou seja, uma solução híbrida.

A base está conectada diretamente ao centro de gerência que irá fazer a requisição de dados para os nós sensores através dos repetidores que estão conectados ao *backhaul* e recebem e retransmitem esta requisição.

Para realizar o acesso aos nós sensores são utilizados dois bytes para endereçamento, sendo um para a identificação das células e outro para o nó sensor. Utilizando este endereçamento é possível realizar uma verificação pelo primeiro rádio do RCH se os pacotes enviados pelo repetidor pertencem ao seu cluster. Se o pacote pertencer a este cluster devem ser enviadas para os nós sensores. Caso o pacote não pertencer a este cluster deverá ser retransmitido para o próximo RCH.

Através deste repetidor, que está na topologia proposta, é possível diminuir o efeito da figura de ruído com o *backhaul* de fibra óptica, pois o mesmo regenera o sinal recebido pelo RoF e aumenta o número de células possíveis ao longo do *backhaul*. Sendo que para o acesso aos nós sensores da RSSF serão utilizados múltiplos saltos e será possível ter uma maior cobertura.

4.1 Análise da cobertura

Na topologia proposta existem redes de sensores sem fio organizadas em células formadas por vários clusters. Cada cluster é formado por um RCH que irá retransmitir os pacotes recebidos pelo repetidor que está ligado ao *backhaul* para os nós sensores.

No cluster da RSSF é importante considerar a área de cobertura, para chegar á distância máxima que o nó sensor poderá estar da antena do rádio transmissor do repetidor. A expressão para o cálculo desta distância será a Equação 7.

Esta equação refere-se ao modelo de *Log-Distance*, que terá seu ambiente caracterizado pelo seu *path loss*.

Através desta equação aplicada após os testes, será possível chegar a distância em relação a uma BER para diferentes taxas da modulação FSK.

A razão pela qual serão analisadas diferentes taxas, está no uso da modulação FSK, também utilizada na 802.15.4 g (IEEE,2006) que permite para a mesma largura de banda ocupada utilizar diferentes taxas devido ao desvio. Para atender a norma da ANATEL (ANATEL, 2008).

Em (ASSUMPÇÃO, 2011) foi utilizada uma taxa fixa e modulação FSK para chegar à distância, considerando 6 cenários distintos. Foi possível identificar que pela topologia montada sem o repetidor houve grande impacto a inclusão do RoF na área de cobertura da RSSF, conforme Figura 13.

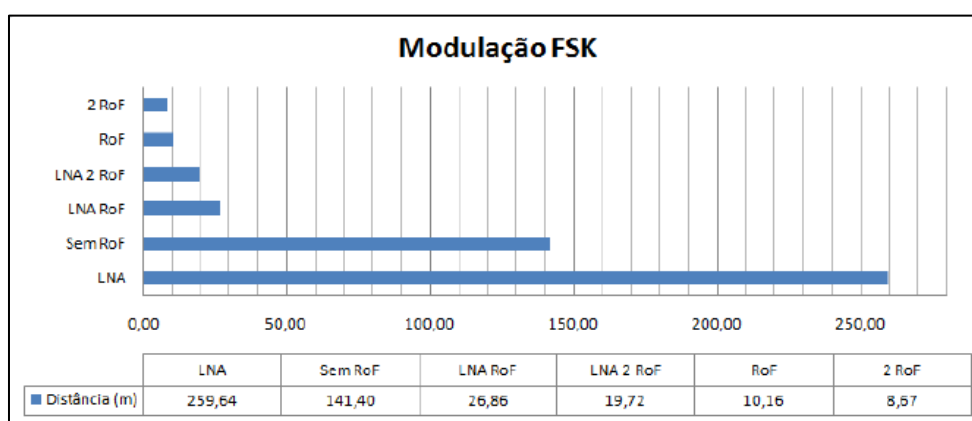


Figura 13: Resultados obtidos para a modulação FSK.(ASSUMPÇÃO, 2011)

Pode-se observar que o amplificador de baixo ruído permite aumentar a distância de cobertura com o RoF.

4.2 Acesso aos nós sensores

A solução proposta neste trabalho para aglutinar os sensores dispersos em uma certa área e enviar as informações para uma centro de gerência, como pode ser visto na Figura 12, é uma topologia híbrida.

Nesta topologia híbrida proposta, o *backhaul* entre os repetidores, a base poderá utilizar rádio ou fibra óptica e a comunicação entre os repetidores e os RCH e o acesso aos nós sensores será através de comunicação sem fio.

A visão de funcionamento desta topologia pode ser verificada na Figura 14, em que os equipamentos utilizados e o caminho de descida e subida percorrido pelo pacote estão demonstrados.

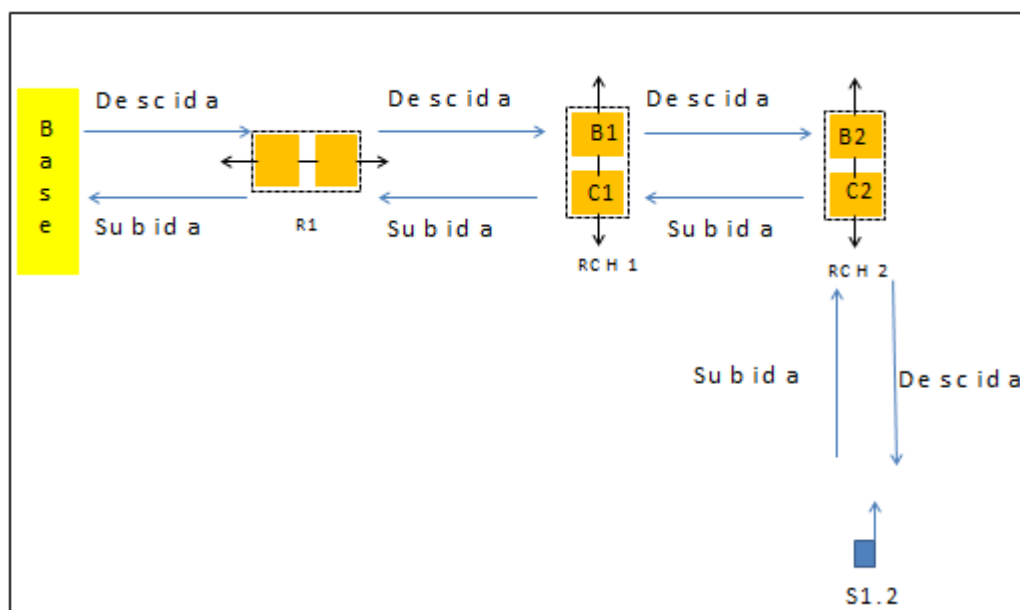


Figura 14: Visão funcional da topologia proposta.

Como pode ser verificado na Figura 14, o pacote é enviado pela base para o repetidor 1, que tem a função de retransmitir todas as informações recebidas através do *backhaul* pela base. Além da retransmissão transparente do pacote, o mesmo regenera o sinal para minimizar a figura de ruído e possibilitar o aumento da área de cobertura das RSSF.

O RCH1 irá receber o pacote pelo nó sensor B1 que irá verificar se no byte de endereçamento de destino refere-se ao id do seu cluster e se foi enviado pela base. Se não for para seu cluster o mesmo irá retransmitir o pacote para o próximo nó sensor B2, que irá verificar se o pacote veio do B1 e se refere ao seu cluster. O B2 irá enviar o pacote pela camada física para o nó sensor C2 e será retransmitido pelo meio sem fio para o nó sensor de destino S1.2, ou seja nó sensor 1 do cluster 2.

No caso exemplificado acima o envio do pacote da base para o nó sensor refere-se ao link de descida, enquanto a resposta do nó sensor para a base ao link de subida.

Em ambos os sentidos, descida e subida, existe apenas um pacote sendo enviado por vez, ou seja, multiplexação por divisão do tempo (TDM), o que evita a colisão do tráfego dos dados (KARL e WILLIG, 2005), caracterizando-se por um sistema centralizado.

Pensando no percurso do pacote, através de uma perspectiva lógica, é possível analisar quais as camadas devem ser implementadas em cada equipamento.

Considerando o mesmo caso exemplificado acima, as camadas que devem ser implementadas estão na Figura 15.

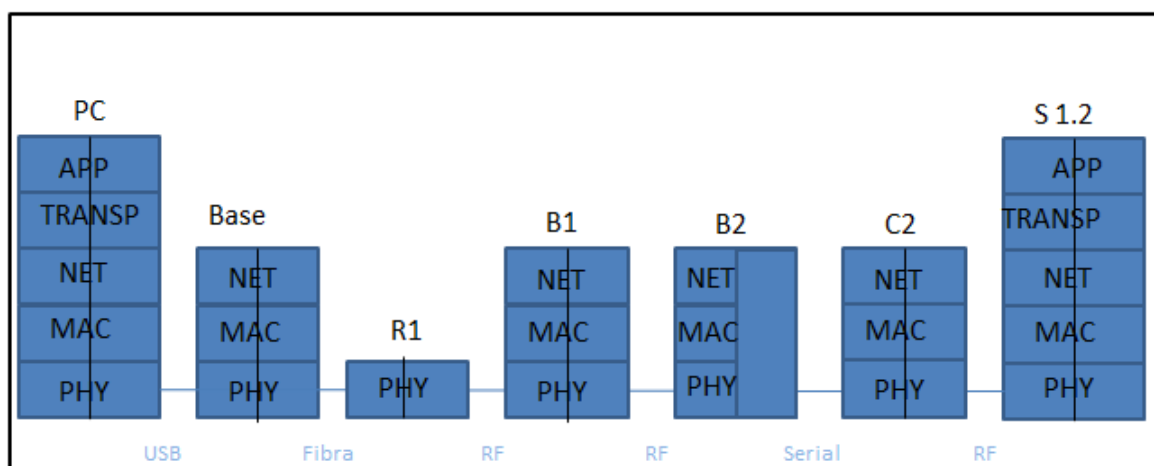


Figura 15: Camadas implementadas na topologia.

Nesta Figura 15 é possível visualizar o computador, que possuiu o software que realiza a gerência desta topologia proposta e irá utilizar as 5 camadas conceituais do modelo TCP/IP descritas (KUROSE e ROSS, 2008):

- PHY- Camada física:
- MAC - Camada de acesso ao meio:
- NET - Camada de rede:
- TRANSP – Camada de transporte:
- APP – Camada de aplicação:

A base está conectada ao computador via usb e serão implementadas as camadas PHY, MAC e NET, TRANSP e APP no computador e até a camada NET na base. Através do *backhaul* será enviado o pacote, que será recebido pelo repetidor 1 e retransmitido, utilizando assim somente a camada PHY. O nó sensor B1 irá receber o pacote e verificar na sua camada NET se o mesmo veio de origem da base e se seu destino é para seu cluster, se não for irá para a camada MAC e depois PHY novamente e será retransmitido para o RCH 2. O B2 irá verificar na camada NET se o pacote veio do RCH1 anterior e se é para seu cluster. Na sequência, irá colocar seu endereço de identificação que está na MAC e através da sua camada PHY irá enviar para o nó sensor C2, que pela sua camada PHY irá receber e retransmitir o pacote para os nós sensores do seu cluster.

O sensor 1 do cluster 2 ao qual o pacote é destinado, irá identificar na camada NET que se refere a ele e irá subir até a camada de aplicação para requisitar os dados medidos pelos transdutores do nó sensor e colocá-los nos bytes corretos do pacote para transmití-lo à base.

A Figura 16 apresenta uma visão temporal da transmissão dos pacotes entre os diversos elementos da rede do exemplo descrito acima.

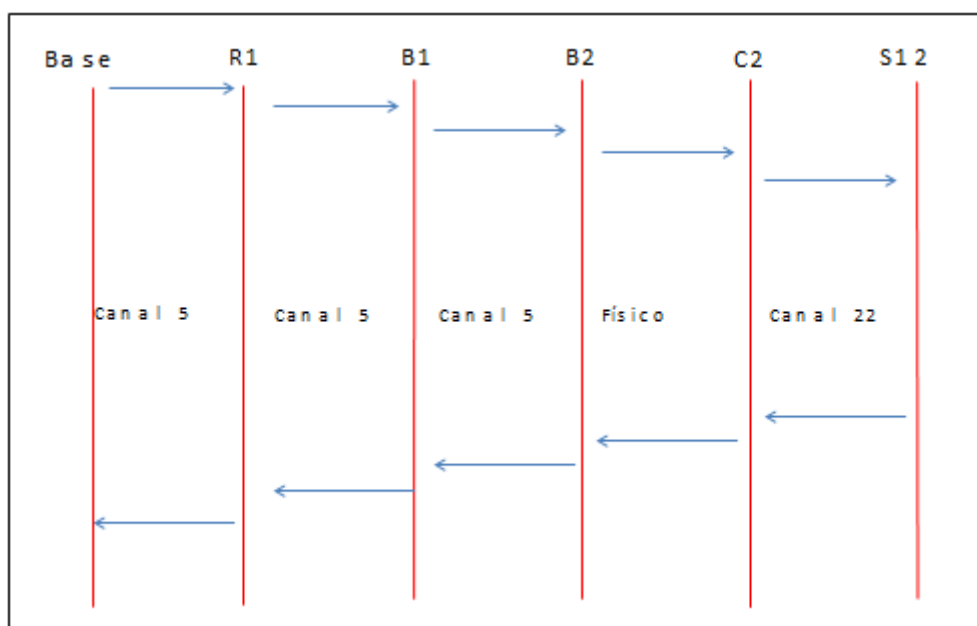


Figura 16: Visão temporal.

Esta visão mostra o envio dos pacotes um por vez, utilizando o sistema de multiplexação por divisão de tempo (TDM), entre os nós sensores até chegar ao seu destino e seu caminho de retorno até a base. As transmissões de dados ocorrem todas na mesma faixa de 915 MHz, sendo que no *backhaul* entre a base ao nó sensor B2 do RCH é utilizado o canal 5, enquanto entre o nó sensor C2 e os nós sensores do seu cluster é utilizado o canal 22.

Estão sendo utilizados dois canais distintos, sendo um para o *backhaul* e outra para acesso do RCH aos nós sensores das RSSF. Desta forma é mitigado o problema de interferência devido ao retorno do diagrama de radiação das antenas, que pode ocorrer entre os clusters de uma mesma célula.

5 MATERIAIS E MÉTODOS

Neste capítulo são apresentados os equipamentos utilizados e a metodologia aplicada na execução dos testes desenvolvidos.

Para a realização do trabalho foi utilizada uma bancada de testes para obter a área de cobertura das RSSF com algumas taxas de transmissão, na faixa de 915 Mhz, utilizando a modulação FSK. Sendo também realizados testes em um *setup* montado no ambiente real para validação da topologia proposta de acesso aos nós sensores.

5.1 Testes em bancada

Foram realizados testes em bancada para emular um canal de forma confinada e assim evitar a interferência dos efeitos de propagação, utilizando caixas blindadas.

O objetivo dos testes é avaliar a distância de cobertura de uma rede RSSF para diversas taxas utilizando a modulação FSK.

Nesta bancada montada em laboratório foram utilizados: base, sensores, RoF, circulador, repetidor, atenuadores e caixa blindada para evitar interferência e confinar o sinal.

Utilizando estes equipamentos com uma metodologia aplicada para realizar as séries de testes, foi possível avaliar o impacto da utilização do repetidor na distância de cobertura de uma RSSF.

5.1.1 Bancada de testes

A bancada de testes foi montada no laboratório LPSira da PUC-Campinas como pode ser visto seu diagrama em blocos na Figura 17.

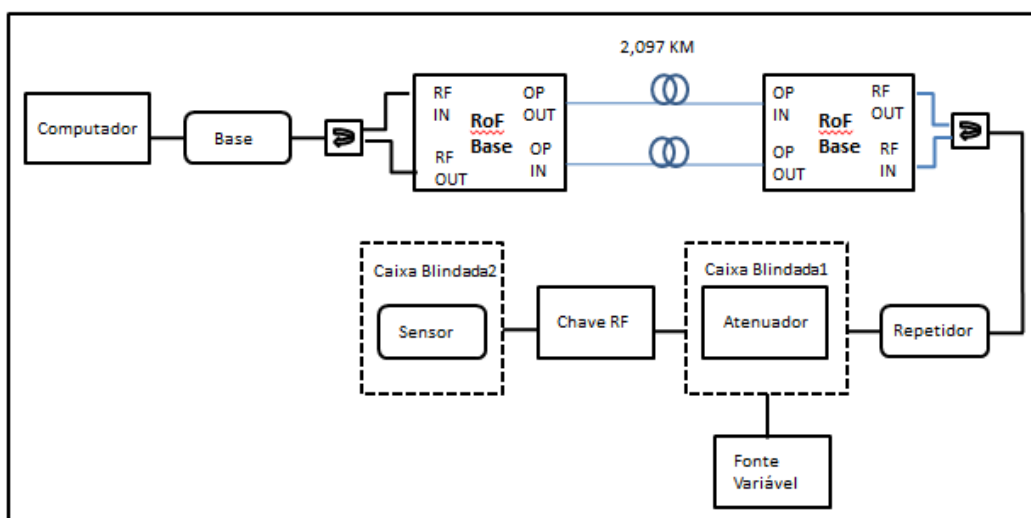


Figura 17: Diagrama em blocos da bancada.

Conforme apresentado na Figura 17, os equipamentos presentes na bancada são:

- Computador: utilizado para executar o software de aquisição dos dados.
- Base e sensor: equipamentos semelhantes que são diferenciados pelo seu endereçamento, firmwares e funções de requisição e envio de dados.
- Circuladores: utilizados para separar o sinal de entrada com o de saída.
- Equipamentos de RoF: são os responsáveis pela conversão do sinal elétrico para o óptico. estão conectados por fibras ópticas.
- Atenuador variável e fonte: responsável por emular diferentes distâncias com a variação em função da tensão aplicada no equipamento.
- Caixas blindadas: utilizadas para realizar o confinamento do sinal de rádio e evitar vazamento do sinal.
- Chave de RF: responsável por atenuar e emular diferentes condições em conjunto com o atenuador.

A bancada de testes de transmissão pode ser vista na Figura 18, onde é possível observar os elementos básicos para o teste.

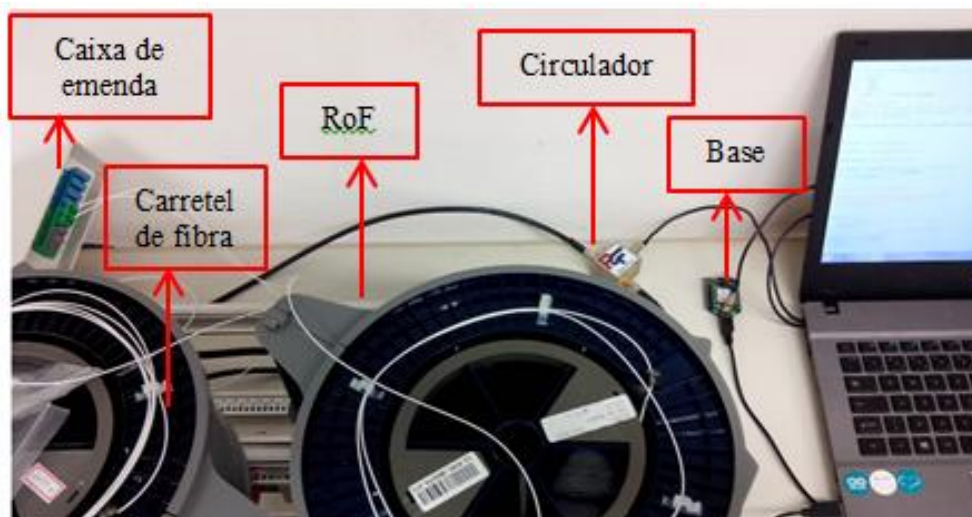


Figura 18: Bancada de transmissão.

A Figura 19 apresenta a bancada de recepção onde temos as caixas blindadas e os equipamentos necessários.

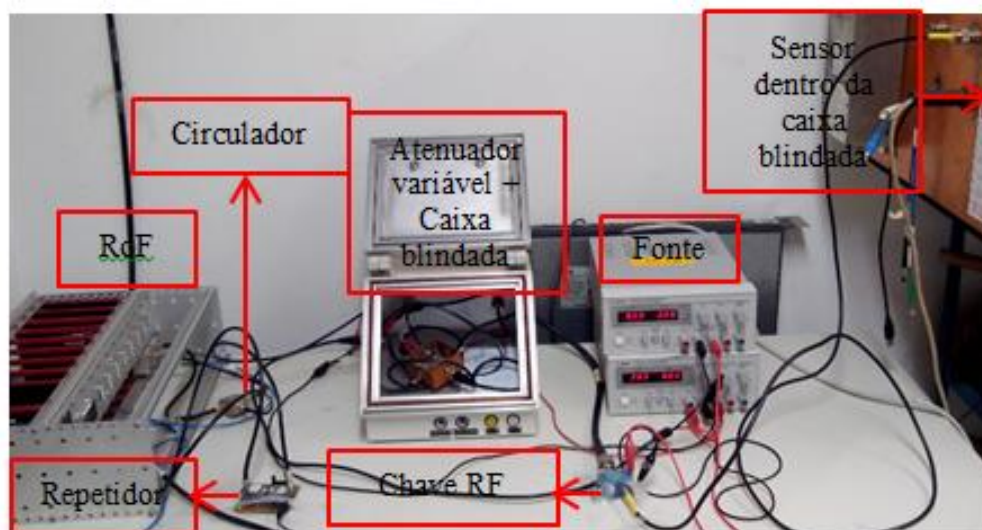


Figura 19: Bancada de recepção.

5.1.2 Emulação do canal

A emulação do canal realizada na bancada foi feita utilizando um atenuador com fonte variável, que ao se excursionar o valor da tensão era variada

à atenuação no canal. Também foi utilizada uma chave de RF para aumentar a atenuação. Sendo que a variação da atenuação simula a variação da distância em uma rede sem fio. O canal utilizado foi cabeado a fim de evitar interferência e ter o sinal confinado e controlado, mantendo a potência de recepção fixa em aproximadamente -92 dBm.

O atenuador utilizado foi RVA -3000 da Mini-circuits (Mini-Circuits) como pode ser verificado na Figura 20 e seu esquemático na Figura 21.

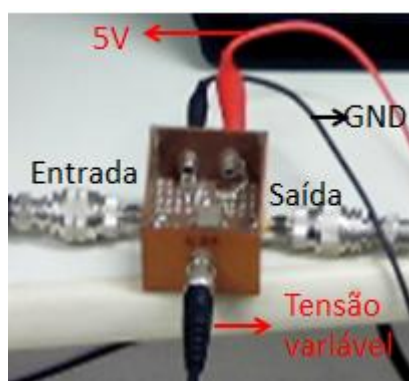


Figura 20: Atenuador utilizado RVA -3000.

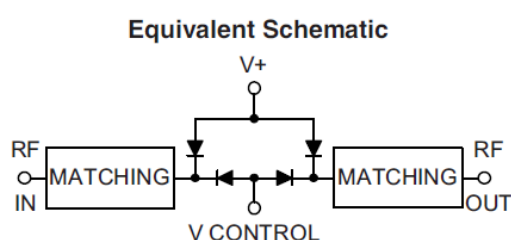


Figura 21: Esquemático equivalente do RVA -3000 (Mini-Circuits).

Foi realizada a calibração do atenuador verificando a tensão aplicada no equipamento e sua atenuação. Esta verificação foi realizada utilizando o gerador de RF para gerar o sinal inserido, as fontes de alimentação, tensão variável, e o analisador de espectro modelo MS2036A para medir a potência do

sinal na saída do atenuador. O experimento realizado pode ser verificado na Figura 22.

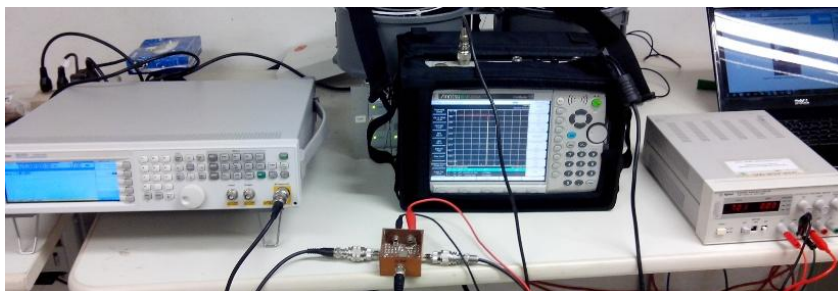


Figura 22: Tensão X Atenuação em 915 MHz.

Desta forma foi montada a curva característica do equipamento, na Figura 23.

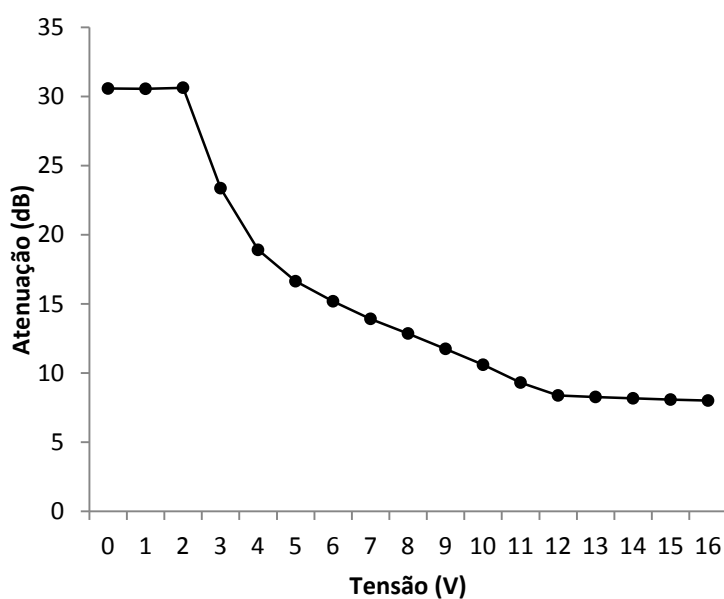


Figura 23: Curva característica medida do atenuador.

Através desta comparação, foi possível confirmar que o atenuador apresenta o seu funcionamento como na sua curva característica, ou seja, quanto menor a tensão aplicada pela fonte maior será a atenuação.

Para a chave de RF, foi utilizado o Modelo ZMSW – 1111 (1111, 2013), mostrado na Figura 24.

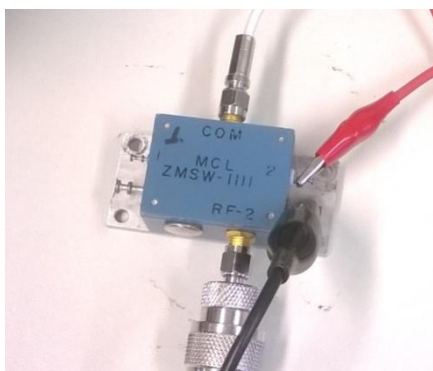


Figura 24: Chave de RF Modelo ZMSW – 1111

Para verificar o funcionamento da chave foi utilizado um gerador de sinal com 915 MHz, frequência utilizada no teste, e na saída um Analisador de Espectro Anritsu modelo MS2036A, capturando os valores necessários, como pode ser verificado o setup na Figura 25.

Com um valor de entrada de -41,55 dBm, no analisador de espectro há um valor de -77 dBm, ou seja, a isolação (diferença) é de 36 dB, isto com a chave fechada e com alimentação de 5 V.

Com um valor de entrada de -31,45 dBm, no analisador de espectro há um valor de -68 dBm, ou seja, a isolação (diferença) é de 37,45 dB, isto com a chave fechada, ou seja, está alimentado com 5 V.

O setup do experimento realizado com a chave de RF pode ser verificado na Figura 25.



Figura 25: Bancada de teste com chave de RF.

Neste *setup* do experimento da chave de RF quando variamos a fonte de alimentação (somente de 0 à 5 Volts, segundo recomendações do fabricante.) quanto mais próxima de zero Volts, maior é a atenuação obtida. Portanto, no cenário de testes, como precisávamos de um valor alto de atenuação, este permaneceu sem alimentação de tensão, atenuando em média 30 dB.

5.1.4 Plataforma de sensores

Para a realização dos testes foram utilizados rádios BE900 com o transceptor CC1101 (INSTRUMENTS, 2010) na plataforma radiuino. Pode ser verificado na Figura 26 os equipamentos utilizados: à esquerda, o nó sensor formado pelo transceptor, micro controlador, transdutor de luminosidade e temperatura e à direita, a base, formada pelo transceptor.

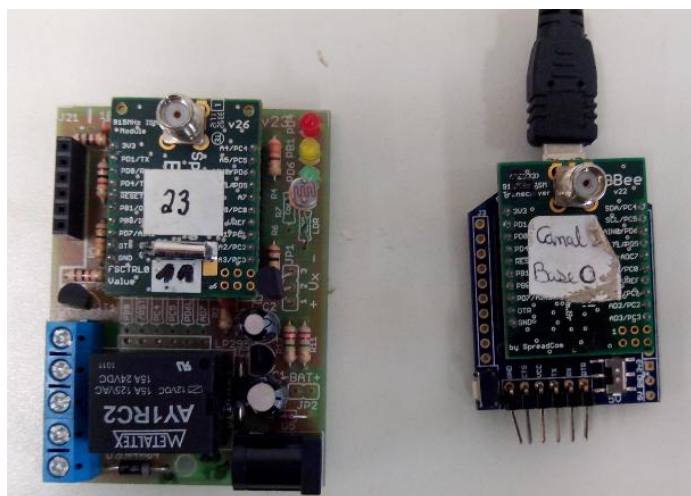


Figura 26: Nó sensor e base.

Estes equipamentos tiveram como principais parâmetros utilizados na configuração dos testes:

- Frequência de operação: 915 MHz.
- Modulação: 2-FSK.
- Potência de transmissão: -30 dBm.
- Potência de recepção: -92 dBm.
- Desvio: 76.1 KHz.
- Banda do filtro: 541,66 KHz.

Foi escolhida a plataforma Rádium (RADIUINO, 2014) em função da sua estrutura, disponibilidade, flexibilidade na alteração dos parâmetros e possibilidade de utilização do mesmo hardware para o nó sensor, base e nó repetidor alterando apenas o firmware.

Sendo que estes nós sensores irão guardar os valores de potência de recepção do link rádio entre os nós sensores obtidos na transmissão de pacotes, como função principal. Porém poderiam estar armazenando e enviando as medidas coletadas no meio através dos transdutores de temperatura, luminosidade ou outra grandeza com o equipamento de aquisição desta grandeza específica, nos pacotes trafegados.

5.1.4.1 Repetidor utilizado

Um dos equipamentos utilizados na bancada de teste foi o repetidor, que tem a função de receber o sinal via rádio BE900, transmitir pela camada física de forma transparente para seu segundo transceptor ou rádio, que irá receber pela serial e transmitir o pacote via rádio com a potência configurada no seu firmware.

Estes dois rádios se comunicam através de uma placa projetada para realizar esta interface serial entre eles, como pode ser visto na Figura 27.



Figura 27: Repetidor utilizado.

5.1.4.2 Aferição dos rádios

Para termos confiabilidade na potência do sinal que estava sendo transmitida, foi realizada a aferição do módulo BE900 para verificar quanto a potência de transmissão.

Foi montado um setup com um computador ligado à base via USB, e a antena do rádio conectada em um cabo com a saída no analisador de espectro Anritsu MS2036A, como pode ser verificado na Figura 28.



Figura 28: Setup para verificar a potência dos rádios.

A potência de transmissão selecionada no firmware deve ser a observada no analisador de espectro.

Foi realizada esta verificação para as potências de -30 dBm, -10 dBm e 10dBm. Este processo também foi realizado para os outros rádios, visto que os mesmos também irão realizar a retransmissão dos dados, e desta forma devem ter os valores próximos do especificado

5.1.5 Pacote utilizado

Para realizar a comunicação entre os nós sensores foi utilizado um pacote com seu tamanho fixo de 52 bytes para seu *payload*. A posição e função de cada byte podem ser vistas na Figura 29 (RADIUINO, 2014).

Cabeçalhos das camadas	PhyHdr(#)				MACHdr(#)				NetHdr(#)				TransHdr(#)					
Posição na variável do cabeçalho	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3		
Função de cada posição de cada cabeçalho	RSSI_D	LQI_D	RSSI_	LQI_UPL		Tempo	Tempo		DST_ID		SRC_ID							
	LINK	LINK	ULINK	INK	Sleep	Sleep1**	Sleep2*	TBD	Recebe	DST_NID	manda	D	COUNT	TBD	TBD	TBD		
Posição dos bytes no pacote	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
Conversores AD	AD0(#)			AD1(#)			AD2(#)			AD3(#)			AD4(#)			AD5(#)		
Posição na variável dos AD	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2
Função de cada posição de cada AD	AD0[0]	AD0[1]	AD0[2]	AD1[0]	AD1[1]	AD1[2]	AD2[0]	AD2[1]	AD2[2]	AD3[0]	AD3[1]	AD3[2]	AD4[0]	AD4[1]	AD4[2]	AD5[0]	AD5[1]	AD5[2]
Posição dos bytes no pacote	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
Entradas e saídas digitais	IO0(#)			IO1(#)			IO2(#)			IO3(#)			IO4(#)			IO5(#)		
Posição na variável dos IO	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2
Função de cada posição de cada IO	IO0[0]	IO0[1]	IO0[2]	IO1[0]	IO1[1]	IO1[2]	IO2[0]	IO2[1]	IO2[2]	IO3[0]	IO3[1]	IO3[2]	IO4[0]	IO4[1]	IO4[2]	IO5[0]	IO5[1]	IO5[2]
Posição dos bytes no pacote	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51

Figura 29: Payload fixo de 52 bytes.

O pacote inteiro possui 64 bytes, sendo utilizados seus bytes como está ilustrado na Figura 30.

Preâmbulo	Sincronismo	Origem	Destino	Payload	CRC
4 bytes	4 bytes	1 byte	1 byte	52 bytes	2 bytes

Figura 30: Pacote utilizado.

Cada campo tem sua função determinada como descrita:

- Preâmbulo: sequência utilizada para indicar o começo da transmissão.

- Sincronismo: este campo tem a função de realizar o sincronismo com o receptor.
- Origem: endereçamento utilizado pelo rádio na sua camada física para indicar a origem do pacote.
- Destino: endereçamento utilizado pelo rádio na sua camada física para indicar o destino do pacote.
- Payload: utilizado como carga útil do pacote, sendo que seus bytes podem ser utilizados como a configuração default indicada na Figura 29, ou pode ser customizado no firmware conforme a necessidade.
 - CRC: checagem e redundância cíclica, mas não foi considerado.

Destes 52 bytes de Payload, estão sendo utilizados os bytes 0 e 2 para a RSSI de *downlink* e *uplink*, os bytes 8 e 10 para ID de destino e origem, como também o byte 12 para contar os pacotes.

5.1.6 Firmware

O firmware é um software embarcado que está gravado no micro controlador. Sendo este responsável pelo funcionamento do nó sensor.

A base, nó sensor e repetidor possuem firmwares diferentes.

No firmware da base, Figura 31, é verificado se existe requisição de pacote, caso exista o mesmo irá transmitir o pacote, aguardar o retorno e verificar se o pacote foi recebido é para a base. Para finalizar é medido a RSSI do rádio.

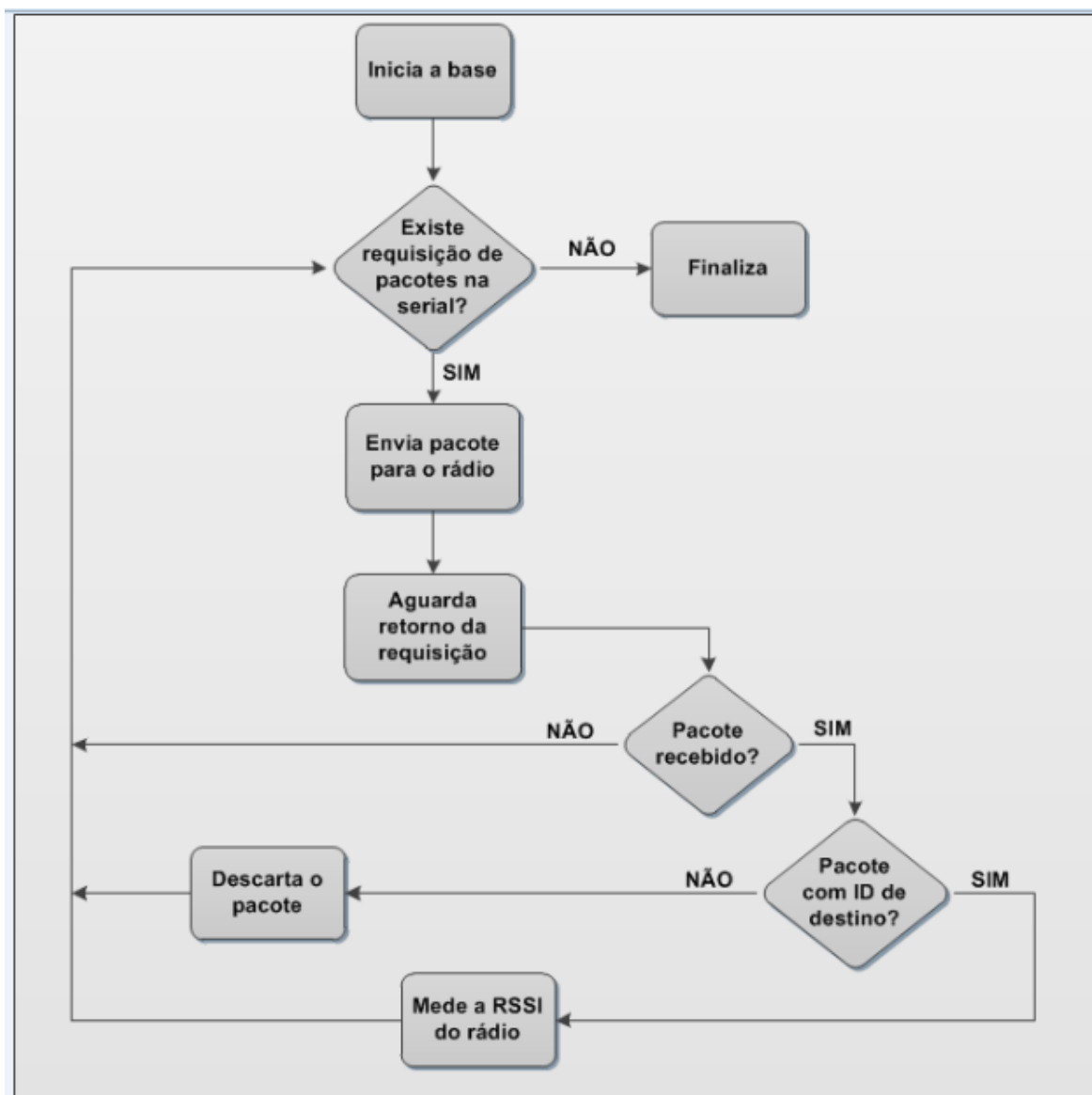


Figura 31: Fluxograma da base.

No firmware do nó sensor, Figura 32, é verificado se existe pacote no rádio, existindo o mesmo requisita o pacote com as informações de RSSI do rádio que são anexadas no pacote. Na sequência o contador é incrementado e o pacote enviado pelo rádio.

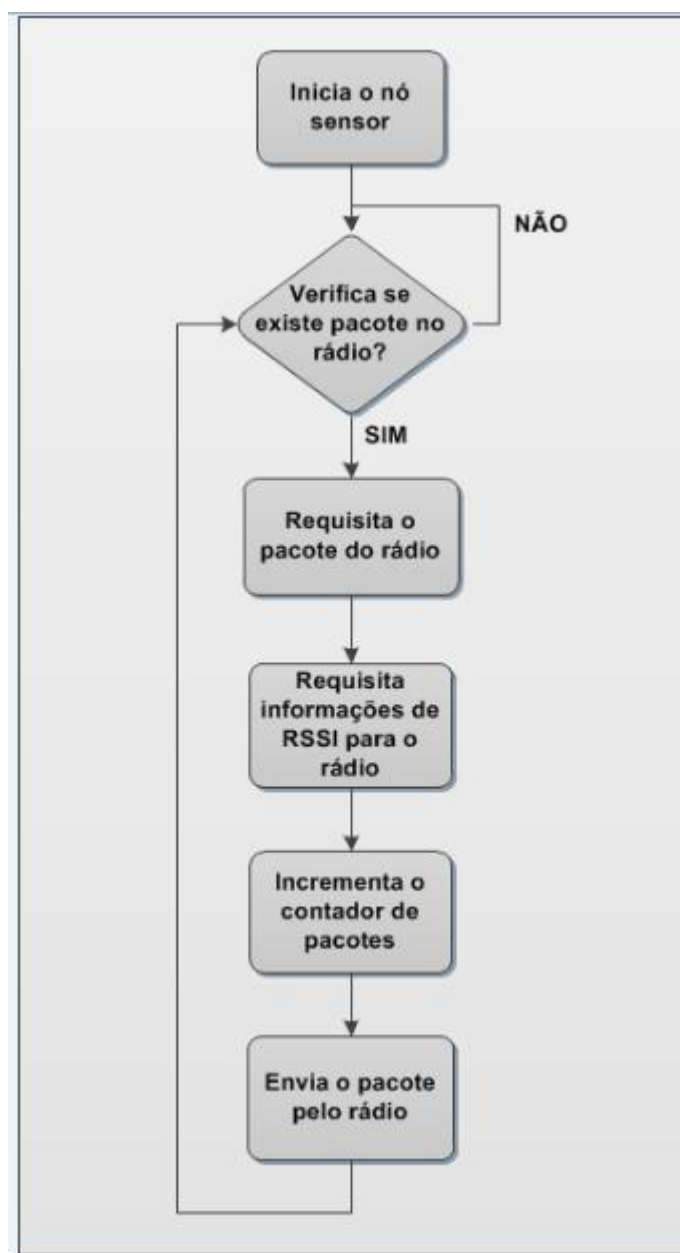


Figura 32: Fluxograma nó sensor.

No firmware do repetidor do nó sensor B, Figura 33, é verificado se existe pacote no rádio, caso exista o mesmo requisita o pacote e as informações de RSSI que serão anexadas no mesmo. Este pacote será enviado pela serial e ficará aguardando.

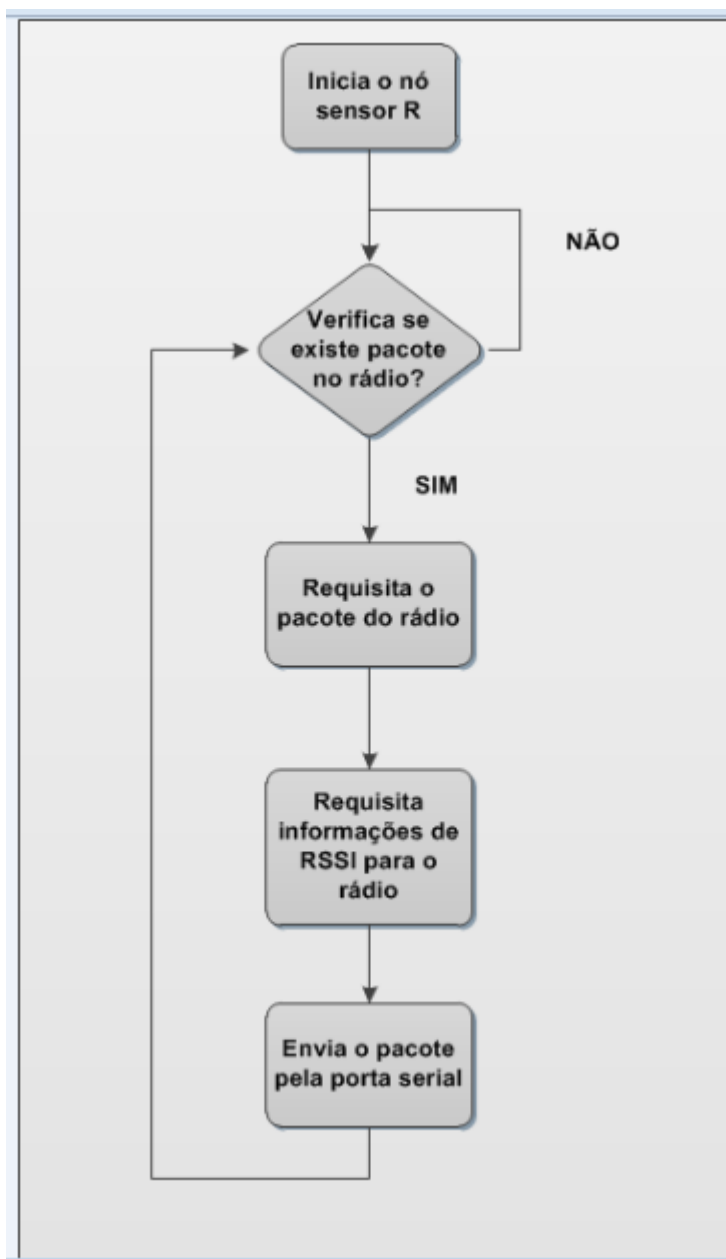


Figura 33: Fluxograma repetidor nó B.

No fluxograma do repetidor do nó sensor C, Figura 34, o mesmo verifica se existe pacote na fila, se existir o pacote é requisitado pela serial e transmitido pelo rádio.

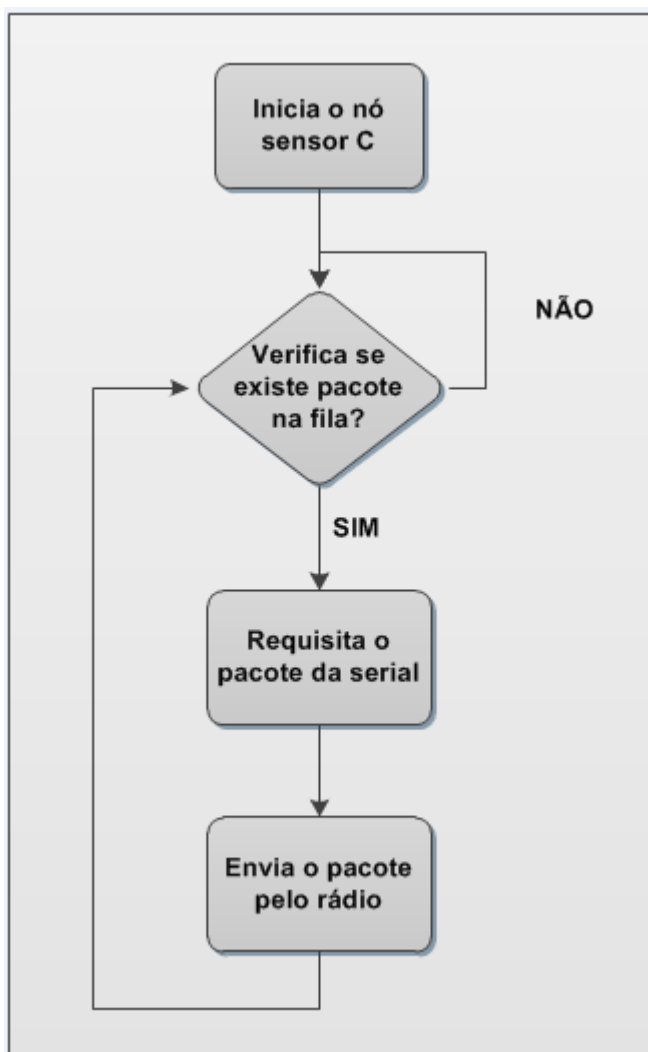


Figura 34: Fluxograma repetidor nó C.

Através destes fluxogramas é possível entender o funcionamento dos firmwares utilizados nos equipamentos da bancada de testes.

5.1.7 Configuração dos registradores

No firmware utilizado no nó sensor é necessário realizar a configuração dos registradores de acordo com os parâmetros de frequência, canal, taxa filtro da banda ocupada, modulação, desvio e potência de transmissão.

Para realizar esta configuração no BE900 foi utilizado o Smart RF Studio, software da Texas do (INSTRUMENTS, 2013). Após aplicar a parametrização, o software gera os valores em hexadecimal para cada registrador

que será exportado e aplicado no firmware, como pode ser visto na parte direita da Figura 35.

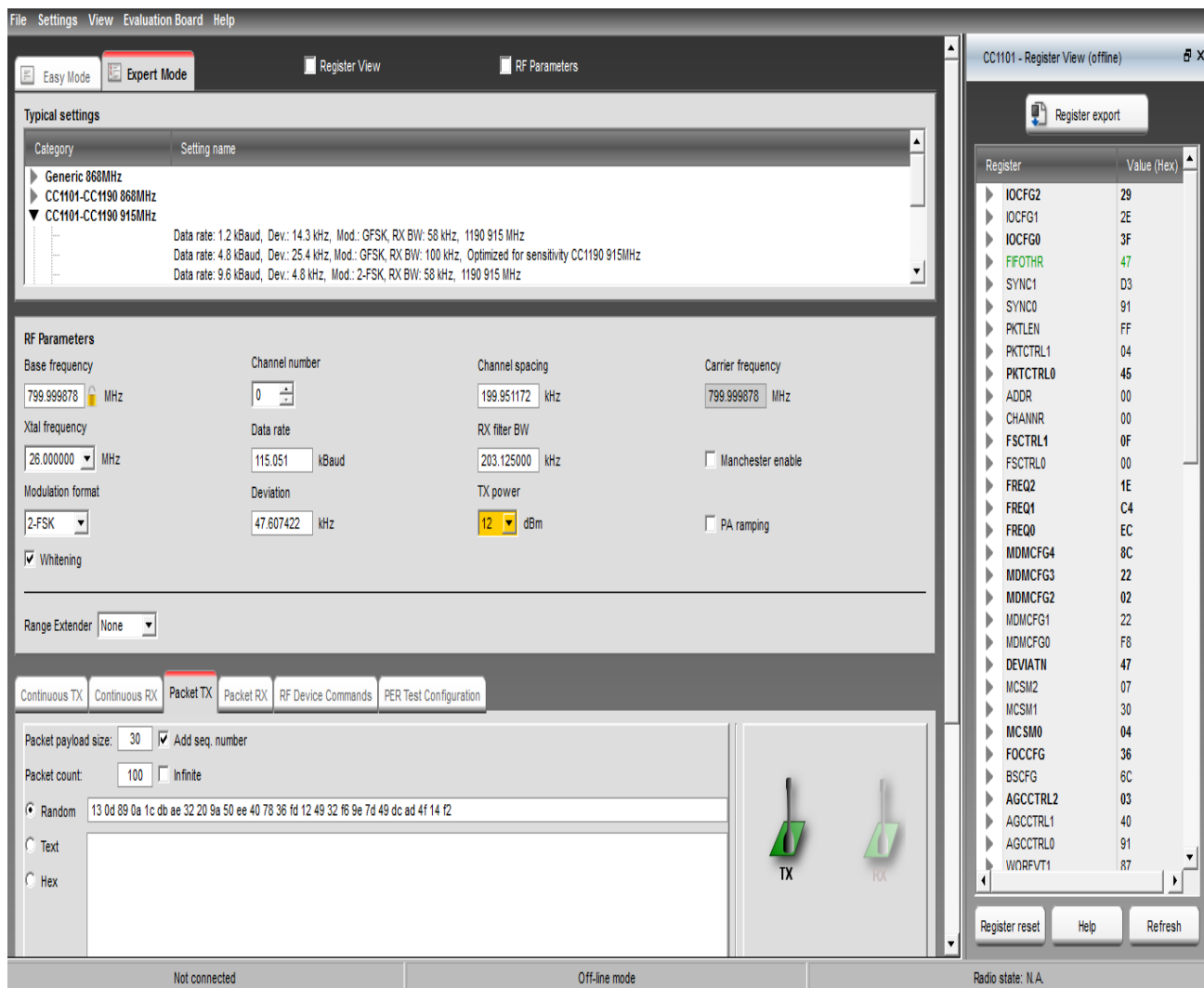


Figura 35: Software Smart RF Studio.

Cada registrador tem uma função específica que pode ser verificada em (INSTRUMENTS, 2010), sendo que tem como exemplo os casos na Figura 36.

Bit	Nome Campo	Início	R/W	Descrição
7		0	R/W	Reservado, escrever 0 para compatibilidade com possíveis extensões futuras.
6	CLOSE_IN_RX[1,0]	0(00)	R/W	0 : TEST1 = 0x31 e TESTE2 = 0x88 1: TEST1 = 0x35 e TESTE2 = 0x81 Observação: As alterações no registo TEST devido a configuração do bit ADC_RETENTION só são vistos internamente na parte analógica. Os valores lidos a partir de registo TEST ao percorrer do modo SLEEP será sempre reinicializado. O bit ADC_RETENTION será definido como 1 antes de entrar em modo de suspensão se o ajuste com uma largura de banda RX for inferior a 325 kHz são procurados no momento do acordar.
5 : 4	CLOSE_IN_RX[]	7(0111)	R/W	Para maiores detalhes veja DN10[8]. Configuração RX Atenuação 0 (00) 0 dB 1(01) 6 dB 2(10) 12 dB 3(11) 18 dB
3 : 0	FIFO_THR[3.0]		R/W	Seta o threshold para o FIFO TX e RX FIFO. O threshold é excedido, quando o número de bytes na memória FIFO é igual ou superior ao valor do threshold. Configuração Bytes em TX FIFO BytesRX FIFO 0(0000) 61 4 1(0001) 57 8 2(0010) 53 12 3(0011) 49 16 4(0100) 45 20 5(0101) 41 24 6(0110) 37 28 7(0111) 33 32 8(1000) 29 36 9(1001) 25 40 10(1010) 21 44 11(1011) 17 48 12(1100) 13 52 13(1101) 9 56 14(1110) 5 60 15(1111) 1 64

Figura 36: Função dos registradores.

5.1.8 Software de coleta

Para realizar a requisição das informações coletadas para o sensor foi utilizado um software programado em linguagem Python (código no Anexo A). Este software é executado em um computador, conectado via USB à base, e solicita as informações escrevendo 52 bytes na serial.

Além de determinar o início e fim dos testes de coleta o software também realiza o processamento dos dados e a análise dos mesmos. A Figura 37 mostra fluxograma que ilustra o funcionamento deste software.

O software de coleta realiza a abertura da serial, do arquivo de log e na sequência monta o pacote padrão e zera os contadores. Este pacote é enviado e aguardado um tempo para verificar se os 52 bytes do mesmo foram recebidos. Se foi recebido o contador de pacotes é incrementado e a resposta é processada assim como o cálculo da RSSI e PER que são exibidos na tela e salvos no log. Este fluxo é realizado até que o número de requisições solicitadas sejam finalizadas.

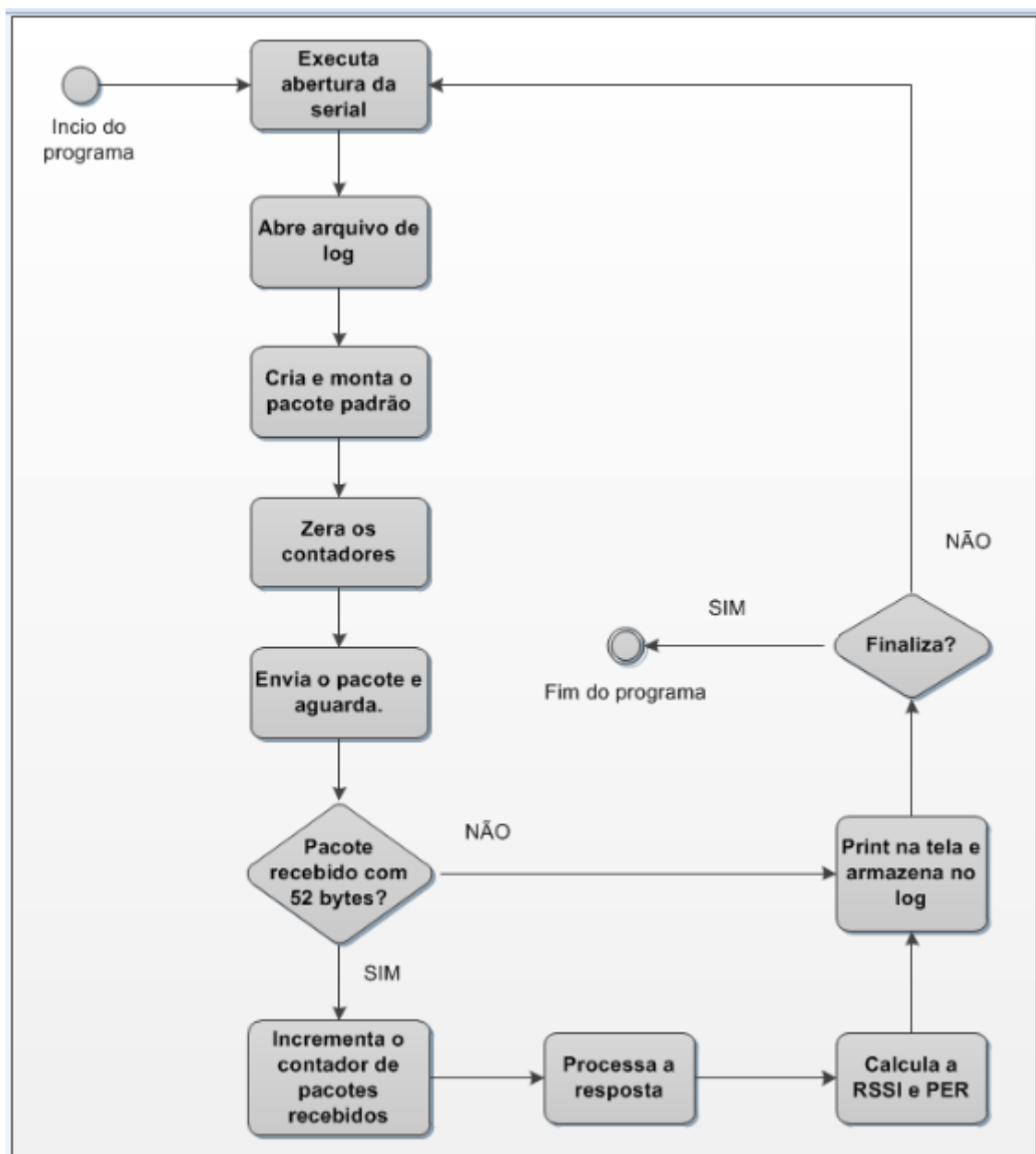


Figura 37: Fluxograma do software de coleta da bancada.

5.1.9 Experimento realizado na bancada

Utilizando a bancada de teste montada no laboratório descrito na seção 5.1.1, foi projetado o experimento para coletar informações de RSSI e perda de pacotes a fim de analisar o impacto na distância dependendo da taxa utilizada para a modulação FSK.

Foram realizadas séries de testes em que foram medidas a RSSI do repetidor para o nó sensor e suas perdas de pacotes, utilizando os equipamentos descritos na seção 5.1.1, com as configurações de firmwares descritas em 5.1.6 e o software de coleta. A partir destes dados foram obtidos pontos que possibilitaram obter a curva característica da modulação FSK para as taxas de transmissão.

Para cada taxa: 4.8; 10; 38.4; 76.8; 99.5 e 150 kbps, de acordo com o documento (INSTRUMENTS, 2013a), foram realizadas cinco séries de testes, em que cada série foi realizada dez mil requisições de informação da RSSI para o nó sensor, sendo este valor suficiente para estabilizar a taxa de erros. (ASSUMPÇÃO, 2011)

Cada requisição foi enviada pela base através do *backhaul* para o nó sensor repetidor B, que através da serial foi transmitido de forma transparente para nó sensor repetidor C, passando pelo atenuador de sinal, chave de rf e pelo seu *backhaul* até chegar ao seu nó sensor. A potência de sinal do pacote recebido pelo nó sensor S é o valor considerado nos testes, assim como a perda de pacotes, para chegar à relação de RSSI com a BER.

5.2 Testes externos

Após o experimento realizado na bancada de teste a fim de obter a distância máxima entre os nós sensores para algumas taxas da modulação FSK, foi realizada na parte externa ao laboratório a implementação da topologia proposta na seção 4.

Os testes externos foram realizados para acessar os nós sensores dos clusters através de múltiplos saltos, e assim validar o funcionamento da solução proposta. Com a utilização do repetidor e RCH nesta topologia proposta foi mitigado o problema da potência do ruído no retorno pelo RoF e aumentada a área de cobertura das RSSF.

5.2.1 Cenário montado

Para realizar os testes da implementação da solução proposta foi montado o cenário conforme mostrado no diagrama da Figura 38.

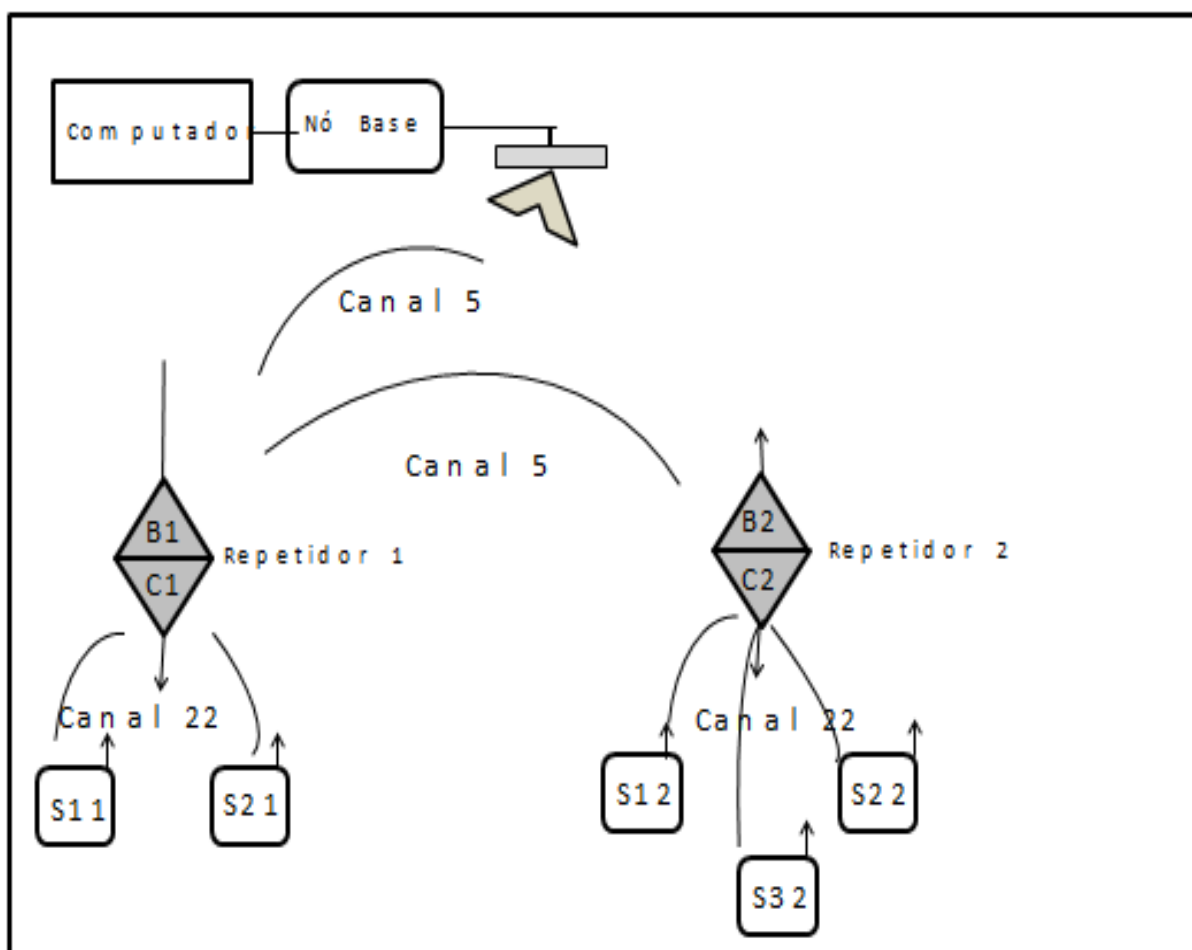


Figura 38: Topologia do teste externo.

Neste cenário o computador está conectado via USB ao nó sensor da base que se conecta a uma antena setorial de 90°. Esta antena irá enviar o pacote na faixa de frequência de 915 MHz, no canal 5 para o nó sensor B1 do repetidor 1, que irá verificar se o pacote é para sua célula, e caso seja irá enviar por serial para o nó sensor C1. O C1 irá transmitir o pacote através de sua antena omnidirecional, no canal 22 para os nós sensores do seu cluster. O nó sensor endereçado irá processar o pacote e transmitir o pacote com a RSSI da potência do sinal recebida. Caso o pacote não fosse para o cluster do repetidor 1, o nó

sensor R1 irá identificar e retransmiti-lo para o repetidor 2 que irá fazer as mesmas verificações do primeiro e retransmitir para o próximo repetidor, sendo desta forma até chegar ao nó sensor correto.

Segue as fotos do ambiente real onde foram realizados os testes de acordo com a Figura 38.



Figura 39: Visão do teste externo.

Na Figura 39 é possível observar a visada entre a antena da base com o RCH1 e a posição do RCH2. Já na Figura 40 é verificada melhor a visada entre o RCH2 e o RCH1.



Figura 40: Visão dos RCH externos.

Os nós sensores que pertencem ao cluster 1 estão dentro do laboratório na parede ao lado do RCH1, e os mesmos podem ser vistos na Figura 41. Enquanto os três nós sensores pertencentes ao cluster 2 estão dentro do laboratório LPSira , ou seja, no 2º andar.

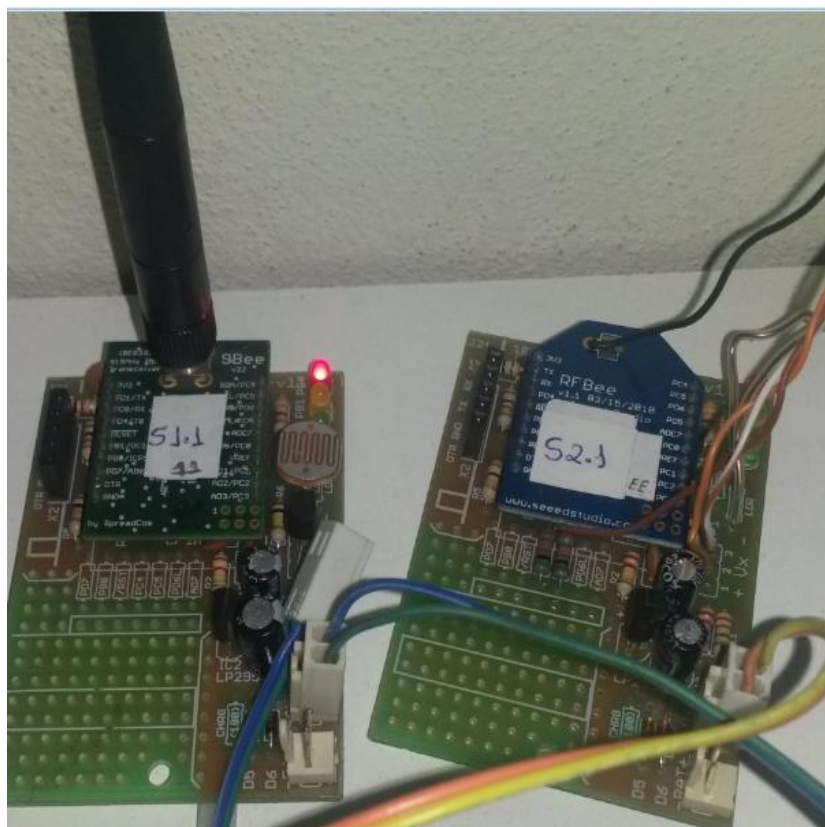


Figura 41: Nós sensores do cluster 1.

5.2.2 Repetidor *cluster head*

Para os testes em ambiente real foram utilizados dois RCH que são semelhantes ao repetidor utilizado na bancada de simulação. Os dois diferem fisicamente pela caixa em que foram colocados para proteger os rádios do tempo, o firmware, e a adição da função de concentrador.

O RCH que foi utilizado nos testes em ambiente real pode ser verificado na Figura 42.



Figura 42: RCH2.

5.2.3 Antenas utilizadas

Como antena para a base foi utilizado um painel de 90°, 915 MHz, setorial, vertical. Sendo que seu *tilt* pode ser ajustado para melhor cobertura, como ilustrada na Figura 43.



Figura 43: Antena painel de 90°.

Para o nó sensor B1 foi utilizada uma antena omnidirecional monopólio vertical ($\lambda/4$) com ganho 2,5 dBi.

Enquanto para os outros nós sensores dos repetidores foi utilizada a antena indoor omni vertical de 2.1dBi (Instrumart, 2013), como pode ser vista na Figura 444.



Figura 44: Antena Omni vertical.

Para o nós sensores dos clusters foram utilizadas somente as antenas do rádio monopólio vertical que vieram de fábrica.

5.2.4 Pacote utilizado

Para os testes da implementação da topologia proposta foi utilizado o mesmo pacote detalhado na seção 5.1.5, com algumas alterações. Dos 52 bytes de Payload, foram utilizados cinco bytes para realizar o endereçamento dos pacotes, seis bytes para armazenar as informações de RSSI uplink e downlink, um byte como contador e um byte para identificar o tipo do pacote que está sendo transmitido.

Na estratégia implementada, este pacote está sendo utilizado para recebimento de informações de RSSI, mas poderiam ser transmitidos pacotes com informações de luminosidade, temperatura, pressão e etc que seriam identificados pelo byte 14 diferente deste pacote.

Como pode ser visto na **Erro! Fonte de referência não encontrada.5**, foram utilizados os bytes 8 e 9 para a identificação do endereço do ID do nó

sensor e da célula de destino. Os bytes 10 e 11 para o ID do nó sensor e do cluster de origem. O cluster de envio do byte 13 possui a informação de qual cluster o pacote passou por último.

S dest	Cel dest	S origem	Cel origem	Cel envio	Tipo pacote
8	9	10	11	13	14

Figura 45: Pacote utilizado.

5.2.5 Firmware

O código dos firmwares dos nós repetidores em (Anexo B), nós sensores em (Anexo C), base em (Anexo E) tiveram algumas alterações na camada de rede em relação ao descrito nos testes em bancada, tais como:

- Incluído o endereçamento da célula a qual o nó sensor pertence.
- Incluído a verificação se o pacote está vindo do seu vizinho ou de outro endereço que não será processado.
- Incluída a verificação no nó repetidor B se o pacote é para sua célula e assim deverá ser encaminhado pela serial para o seu nó sensor C, ou se refere à outra célula e deverá ser retransmitido para o próximo repetidor.
- Se byte 9 (cluster de destino) = Net.cel_addr (endereço da célula) and byte 13 = Net.cel_addr -1

O pseudo código utilizado no firmware dos RCH na camada de rede:

- B1 do RCH1 recebe o pacote
 - Verdadeiro: Escreve no byte 13 o seu Net.cel_addr e envia as informações para o repetidor C1 que irá transmitir através de sua antena o pacote

- Falso: Irá colocar o seu endereço no byte 13 do pacote e irá transmitir o pacote

O próximo repetidor irá receber este pacote e irá realizar as mesmas verificações feitas pelo anterior. Isto será realizado até que o pacote chegue ao seu nó sensor de destino.

Este nó sensor que recebeu o pacote irá montar o seu pacote com suas informações de RSSI e outras grandezas, como por exemplo, temperatura e umidade, que podem estar configuradas no seu firmware.

Neste pacote os endereços de bytes de origem 10 e 11 e de destino 8 e 9, serão invertidos.

Na volta o repetidor C irá verificar:

- Se o endereço do byte 11(célula de origem)= cluster do RCH
 - Verdade: O pacote será enviado para o repetidor B que irá transmitir por sua antena
 - Falso: O pacote não será enviado.
- Se outro repetidor B receber o pacote será verificado:
- Se o byte 9 (célula de destino) < Net.cel_addr e byte 13 = Net.cel_addr + 1
 - Verdade: Escreve no byte 13 e retransmite o pacote.
 - Falso: O pacote não será retransmitido.

5.2.6 Software de coleta dos testes externos

Assim como na bancada, nos testes externos foi utilizado um software programado em linguagem Python, para a aquisição e análise dos dados, está no (Anexo F). O seu funcionamento pode ser entendido através do fluxograma da Figura 46, em que é feita a entrada dos parâmetros do teste e o tempo de execução do mesmo. Na sequência é aberta a porta serial que percorre todos os

clusters da célula e todos os nós sensores enviando o pacote de dados, um por vez. Aguarda o recebimento do pacote e verifica se foi recebido corretamente todos os bytes. Se tiver sido recebido corretamente será calculada a RSSI e a PER e armazenados no log, caso contrário irá exibir uma mensagem de erro e gravar as informações no log. Será verificado se já foram percorridos todos os sensores de todos os clusters da célula. Este fluxo será executado novamente até que o tempo colocado na entrada seja atingido.

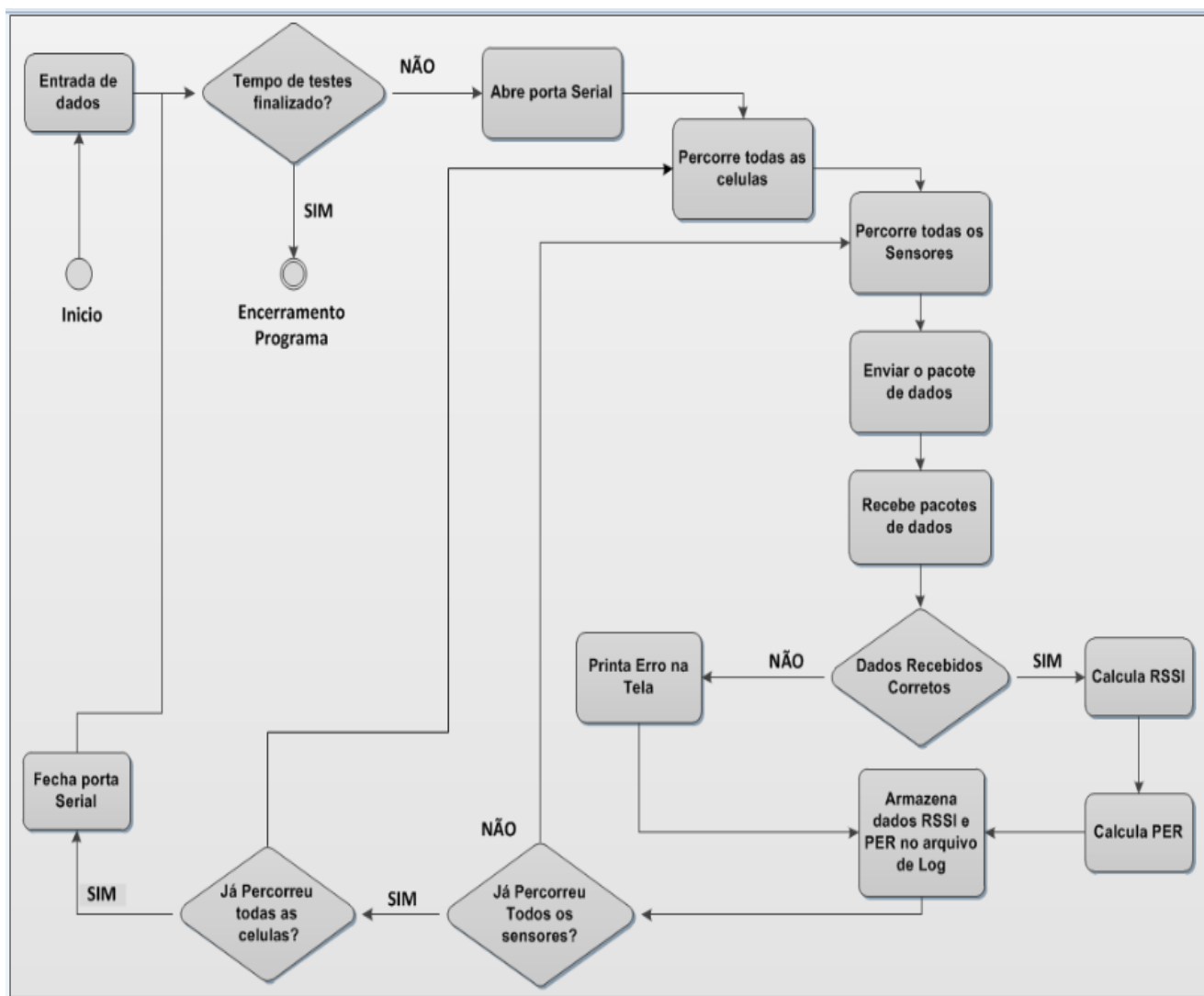


Figura 46: Fluxograma do software de coleta do teste externo.

5.2.7 Experimento realizado

Foram realizadas séries de ensaios para validar o correto funcionamento da solução implementada. Neste experimento foram utilizados 2 clusters, sendo que o primeiro com dois nós sensores e o segundo com 3 nós sensores.

Nestas séries foram medidas as RSSI's entre os nós sensores e a PER entre a base e os nós sensores. Cada série foi realizada durante o período de 24 horas realizando requisição por vez para cada nó sensor, passando por todos os nós sensores para completar um ciclo.

Foram utilizados como parâmetros:

- Taxa de transmissão: 4,8 kbps.
- Modulação: 2-FSK.
- Potência de transmissão: 10 dBm.
- Faixa de frequência: 915 MHz.

Foram realizados testes para validar que o pacote enviado para o nó sensor do RCH 2 estava seguindo o caminho correto, ou seja, indo da base para o R1 e sendo retransmitido para o B2, que irá enviar o pacote para o C2 e retransmitir para o um nó sensor pertencente ao seu cluster.

6 ANÁLISE DE CENÁRIOS DE APLICAÇÃO

A topologia proposta neste trabalho está na Figura 12, sendo que considerando o *backhaul* de fibra óptica entre os repetidores e a base, será feita a análise do número de células possíveis para um determinado cenário. Conforme a estrutura da Figura 47.

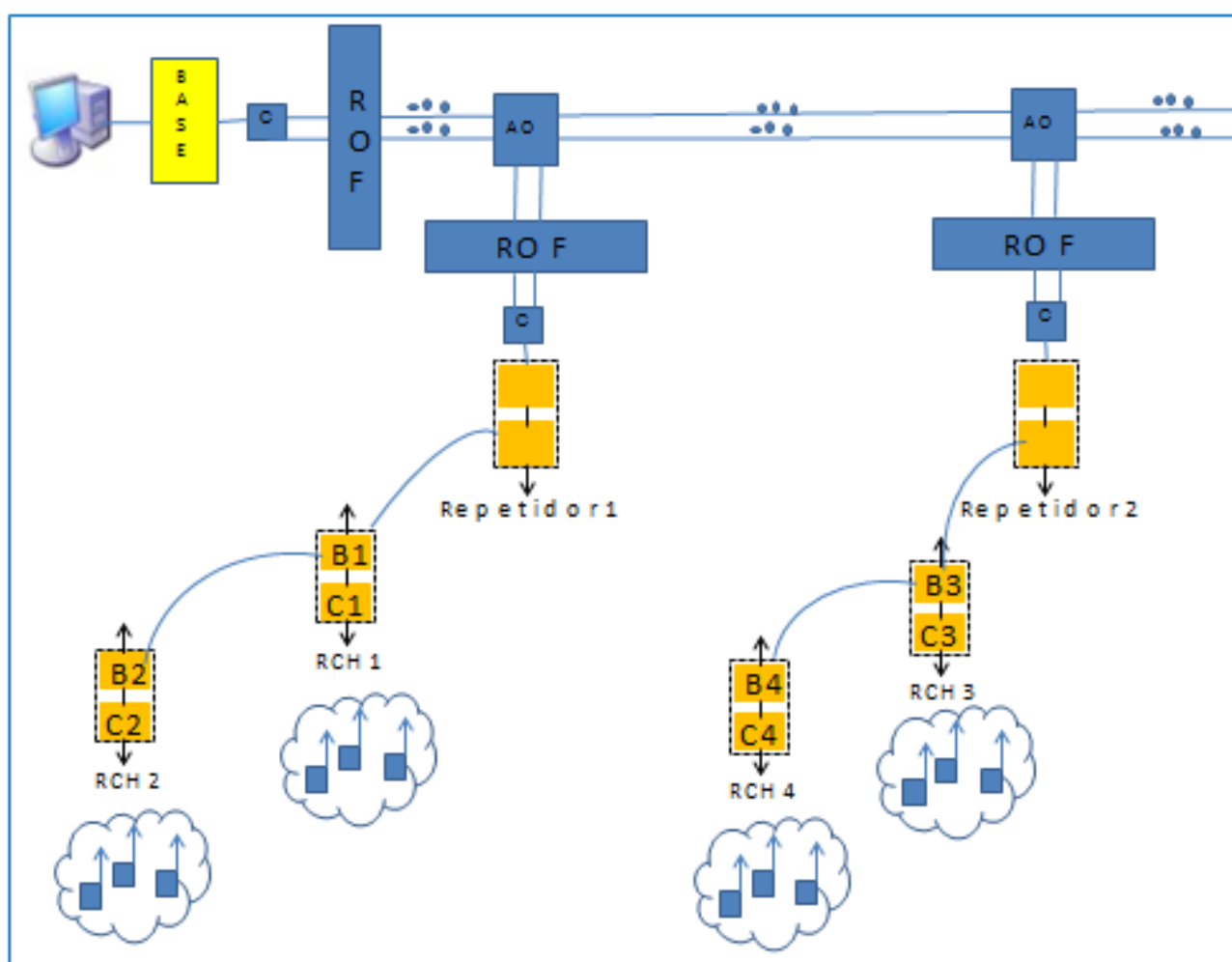


Figura 47: Topologia com *backhaul* de fibra óptica.

Através desta Figura é possível verificar os equipamentos e a forma que é montado o *backhaul* de fibra óptica entre os repetidores e a base. Sendo que cada elemento é resumidamente descrito a seguir:

- Base: Está conectada ao centro de gerência através de uma serial.
- Circulador: Utilizado para separar o sinal de saída e de entrada do rádio, e para evitar a realimentação do equipamento.
- RoF: Usado para interligar o centro de gerência com as células, onde temos os repetidores, interligados via fibra óptica.
- Acoplador óptico: Dispositivo utilizado para acoplamento de fibras ópticas, para realizar uma derivação nos pontos onde estão as células.
- Repetidor: É composto de dois rádios ligados através de uma serial, tendo por objetivo melhorar a cobertura rádio das células através da recepção e retransmissão do pacote na entrada do RoF. Com esta configuração também é possível fazer uma interface rádio com a fibra em um canal e em outro canal para se comunicarem com os sensores.
- RCH (Repetidor *cluster head*): É o mesmo hardware do repetidor, porém além da função de repetidor o mesmo desempenha a função de cluster head. Sendo que também realiza a verificação se o pacote é endereçado ao seu cluster ou não para desempenhar suas funções.

6.1 Exemplo Aplicado

Para analisar a topologia proposta foi considerado como exemplo um cenário ilustrativo de uma rede subterrânea de uma empresa de distribuição de energia elétrica do estado de São Paulo. Neste cenário com a extensão de 18 km de linha de transmissão existem 50 transformadores e 200 chaves seccionadoras que podem ter suas grandezas de temperatura, umidade, pressão monitoradas para a detecção de uma possível alteração ou falha dos equipamentos, ao longo de uma linha radial de energia.

Para este cenário proposto serão considerados os seguintes parâmetros:

- Modulação: FSK, para atender a banda ocupada requerida pela regulamentação.
- Faixa ocupada: 541,66 KHz.
- Frequência de operação: 915 MHz.
- SNR: 8,3dB para uma PER de 5% (ASSUMPÇÃO, 2011).
- EIN: -134 dBm/Hz, conforme indicado no datasheet (Zonu, 2010).

Para determinar o número de células de nós sensores possíveis é necessário calcular a potência de recepção, ou sensibilidade da base através da Equação 19 e comparar com o valor obtido da potência de recepção de uplink da Equação 20, utilizando os parâmetros acima.

Através das considerações feitas e dos cálculos realizados pelas equações demonstradas, verifica-se que é possível ter até 11 células de sensores para o monitoramento dos transformadores ao longo dos 18 km da rede subterrânea proposta, conforme Tabela 4.

Tabela 4: Resultados obtidos.

n (número de células WSN)	Sensibilidade na Base (dBm)	Prx <i>uplink</i> –volta (dBm)
5	-61,40	-21,2
10	-58,37	-51,2
11	-57,97	-57,2
12	-57,58	-63,2

Através da Tabela 4 observa-se que a potência de volta com 11 células iguala-se à potência de ida, sendo este o limite de células. Desta forma haverá uma célula a cada 1,63 km, monitorando cada uma até 5 transformadores em sua área de cobertura da rede RSSF.

Não será considerado o atraso na fibra, pois no projeto apresentado não foi considerado o trabalho da MAC.

7 RESULTADOS

Neste capítulo serão apresentados os resultados referentes à bancada de simulação e dos testes externos da implementação da topologia de acesso aos nós sensores.

7.1 Resultados dos testes da bancada

Nos testes realizados na bancada de simulação foram obtidos a BER para cada taxa utilizada da modulação FSK, fixando a potência de recepção em torno de -92 dBm.

Para obter estes valores que compõem o gráfico da Figura 48 foi necessário realizar a diferença do E_b/N_0 em relação à Prx, sendo que este valor obtido para cada taxa foi replicado para termos assim a RSSI para cada E_b/N_0 correspondente. Com estes valores foi possível montar a curva de Prx por BER para cada taxa como na Figura 48, considerando como base a curva característica E_b/N_0 pela probabilidade de erro (P_e) da Figura 9.

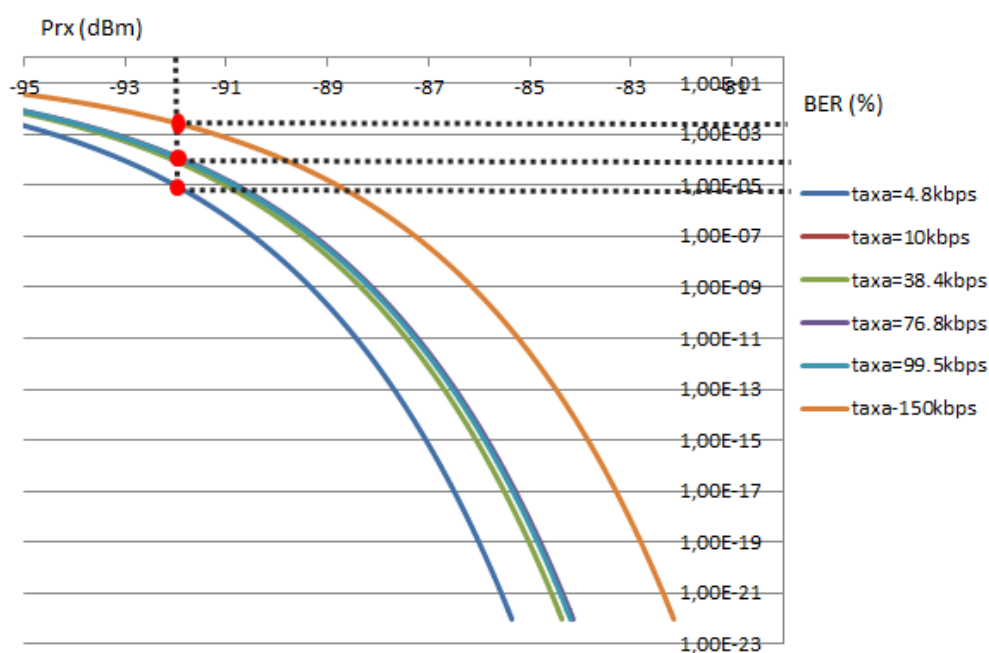


Figura 48: Gráfico Prx X BER do resultado dos testes em bancada para a modulação FSK.

Através deste gráfico é possível verificar que quanto maior a taxa utilizada maior é a BER correspondente, considerando o mesmo valor da Prx. Porém para as taxas intermediárias 38.4, 76.8 e 99.5 kbps os valores obtidos foram bem próximos da relação sensibilidade (Prx X BER).

Com estes valores da potência de recepção obtidos e sua BER correspondente, é possível calcular a distância alcançada, como indicado na Equação 8, em função da atenuação dos ambientes e para cada taxa que foram realizados os testes. Estes resultados para cada taxa podem ser verificados na **Erro! Fonte de referência não encontrada..**

BER (%)	Taxa (Kbps)	S (dBm)	d (m)			
			$\beta=2$	$\beta=3$	$\beta=3,41$	$\beta=4$
9,99E-06	4,8kbps	-92	32,85	10,26	7,75	5,73
	10kbps	-92,1	33,23	10,34	7,81	5,76
	38.4kbps	-91,01	29,31	9,51	7,25	5,41
	76.8kbps	-90,79	28,58	9,35	7,14	5,35
	99,5kbps	-90,84	28,74	9,38	7,17	5,36
	150kbps	-88,8	22,72	8,02	6,25	4,77
8,98E-05	4,8kbps	-92,99	36,81	11,07	8,29	6,07
	10kbps	-93,09	37,24	11,15	8,34	6,10
	38.4kbps	-92	32,85	10,26	7,75	5,73
	76.8kbps	-91,78	32,03	10,08	7,64	5,66
	99,5kbps	-91,83	32,21	10,12	7,66	5,68
	150kbps	-89,79	25,47	8,66	6,68	5,05
1,36E-04	4,8kbps	-93,21	37,76	11,25	8,41	6,14
	10kbps	-93,31	38,19	11,34	8,47	6,18
	38.4kbps	-92,22	33,69	10,43	7,87	5,80
	76.8kbps	-92	32,85	10,26	7,75	5,73
	99,5kbps	-92,05	33,04	10,30	7,78	5,75
	150kbps	-90,01	26,12	8,80	6,78	5,11
1,73E-04	4,8kbps	-93,34	38,33	11,37	8,49	6,19
	10kbps	-93,44	38,77	11,46	8,54	6,23
	38.4kbps	-92,35	34,20	10,54	7,94	5,85
	76.8kbps	-92,13	33,34	10,36	7,82	5,77
	99,5kbps	-92,18	33,54	10,40	7,85	5,79
	150kbps	-90,14	26,52	8,89	6,84	5,15
2,79E-03	4,8kbps	-95,2	47,48	13,11	9,62	6,89
	10kbps	-95,3	48,03	13,21	9,69	6,93
	38.4kbps	-94,21	42,36	12,15	9,00	6,51

	76.8kbps	-93,99	41,30	11,95	8,87	6,43
	99,5kbps	-94,04	41,54	11,99	8,90	6,45
	150kbps	-92,00	32,85	10,26	7,75	5,73

Figura 49: Distâncias alcançadas em relação as BER's obtidas para as taxas testadas.

Através destes resultados, é possível observar que para as taxas 4.8 e 10 kbps os valores de distâncias são próximos. Para as taxas intermediárias 38.4 , 76.8 e 99.5 kbps o alcance diminui mas é pouco expressivo a diferença entre elas. Ao se comparar a menor taxa de 4.8 com a maior de 150 kbps é possível perceber uma diferença real entre as distâncias alcançadas para todos os *path loss* calculados, conforme o esperado.

Os resultados obtidos na banca são coerentes com o esperado de quanto maior a taxa utilizada menor a distância alcançada, porém podem não ter sido tão expressivos, pois nos pacotes foram enviados os valores da RSSI apenas, e não grandezas coletadas através dos transdutores dos nós sensores.

7.2 Resultados dos testes no ambiente externo

Para os ensaios realizados em ambiente externo, foi possível observar o funcionamento da topologia para acessar os nós sensores através do firmware implementado para múltiplos saltos de uma RSSF fixa e utilizando a taxa de 4.8kbps. Através destes ensaios obtivemos os valores de RSSI e PER para cada dia de coleta realizada.

Os resultados obtidos para os quatro dias de testes realizados para os dois nós sensores do cluster 1, S1.1 e S2.1 e para os três nós sensores do cluster 2, S1.2, S2.2 e S3.2, foram as Tabela 5 e Tabela 6. Nestas tabelas são apresentados os valores médios diários de PER e potência de recepção de uplink e downlink entre:

- Nó sensor C do RCH e o nó sensor S.
- Base e nó sensor B do RCH
- Nó sensor B do RCH e seu próximo nó sensor B do RCH.

Tabela 5: Resultados dos nós sensores do cluster 1.

Data	RSSI (dBm)				PER (%)
	C1 S1.1	S1.1 C1	B1Base	Base B1	
					S1.1
10-03-15	-58,13	-54,91	-82,18	-83,12	0,4149
11-03-15	-56,09	-53,12	-80,62	-81,36	3,33
12-03-15	-57,54	-54,51	-81,21	-82,24	0,84
13-03-15	-53,26	-50,56	-76,14	-77,15	6,6
	C1 S2.1	S2.1C1	B1Base	Base B1	S2.1
10-03-15	-66,25	-67,67	-82	-82,95	6,2
11-03-15	-64,14	-65,50	-82,00	-82,75	1,5
12-03-15	-66,40	-67,76	-81,34	-82,37	0,70
13-03-15	-65,36	-66,95	-78,40	-79,45	3,8

Através destes resultados dos nós sensores do cluster 1 é possível verificar que o protocolo funcionou adequadamente e os valores de PER ficaram dentro do esperado para as redes de sensores sem fio de 5%, exceto no dia 13-03-15 para o sensor S1.1 que devido as chuvas intensas deve ter provocado esta PER mais elevada de 6,6 %. Sendo que em muitos casos podem ser suportadas taxas de erros maiores (KARL e WILLIG, 2005).

Na Tabela 6 é possível verificar os resultados obtidos para o cluster 2.

Tabela 6: Resultados dos nós sensores do cluster 2.

Data	RSSI (dBm)						PER (%)
	C2 S1.2	S1.2 C2	B1Base	Base B1	B1B2	B2B1	
10-03-15	-65,94	-67,26	-81,66	-82,58	-55,53	-55,15	1,057
11-03-15	-61,61	-62,82	-77,26	-77,98	-52,61	-52,09	7,3
12-03-15	-62,08	-63,41	-75,13	-76,06	-53,15	-52,62	8,2
13-03-15	-40,51	-41,25	-49,31	-49,97	-34,43	-34,14	39,59
	C2 S2.2	S2.2 C2	B1Base	Base B1	B1B2	B2B1	S2.2
10-03-15	-69,92	-72,75	-81,75	-82,68	-55,61	-55,23	0,93
11-03-15	-64,22	-66,73	-74,58	-75,27	-50,75	-50,23	10,7
12-03-15	-68,97	-72,02	-81,21	-82,19	-57,42	-56,86	0,89
13-03-15	-67,83	-70,72	-79,24	-80,30	-55,46	-54,94	2,8
	C2 S3.2	S3.2 C2	B1Base	Base B1	B1B2	B2B1	S3.2
10-03-15	-65,44	-65,85	-82,03	-82,98	-55,79	-55,41	0,60
11-03-15	-61,31	-61,70	-76,50	-77,21	-52,07	-51,55	8,3
12-03-15	-64,51	-64,86	-81,24	-82,22	-57,45	-56,89	0,87
13-03-15	-64,95	-65,36	-79,77	-80,82	-55,83	-55,31	2,2

Através da Tabela 6 é possível verificar que os resultados do cluster 2 foram semelhantes ao cluster 1. Sendo que no dia 13-03-15 o sensor S1.2 obteve uma PER de 39,5%, podendo ser considerada efeito das chuvas intensas, assim

como o S1.1 teve uma PER maior que as outras médias diárias. É possível observar que no cluster 2 a média da PER foi mais elevada que no cluster 1, o que é esperado devido ao maior número de saltos para acessar o nó sensor.

Com estes resultados ficou comprovado que o protocolo implementado no ambiente externo funcionou adequadamente.

8 CONCLUSÃO

Este trabalho fez a proposta e avaliação de uma topologia híbrida utilizando rádio ou rádio sobre fibra como backhaul, e realizando o acesso aos nós sensores através de multi-saltos. A fim de avaliar a área de cobertura ponto a ponto em uma rede de sensores sem fio foram realizados testes em uma bancada de emulação, e para avaliar o funcionamento da topologia proposta de forma a acessar aos nós sensores através de múltiplos-saltos foram realizados testes em um ambiente externo.

Nos testes realizados em bancada utilizando o *backhaul* de fibra óptica com o repetidor, foi possível observar que para as taxas de 38.4, 76.8 e 99.5 kbps as distâncias foram bem próximas ao se diminuir a BER. O comportamento esperado de quanto maior a taxa pior a BER e menor a distância alcançada foi obtido entre a menor taxa de 4,8 kbps e a maior taxa de 150 kbps, enquanto para as taxas intermediárias os valores de BER e distância foram bem próximos.

No ambiente externo foi possível observar através dos testes que a PER esperada de 5% para as RSSF foi obtida para o cluster 1, porém com dois pontos fora da média. Para o cluster 2 tivemos 5 valores de PER acima do esperado e dentre estes um considerado como anomalia, provavelmente devido as chuvas intensas nos dias dos testes. Este número de casos fora da média da PER no cluster 2 é esperado, pois para acessar o nó sensor temos um salto a mais em relação ao cluster 1.

Através dos testes realizados na bancada e no ambiente real é possível comprovar o funcionamento da topologia híbrida proposta neste trabalho, de forma a aglutinar os nós sensores e enviar as informações para o centro de gerência.

8.1. Contribuições

Este trabalho teve como contribuição a proposta de uma topologia de acesso aos nós sensores por múltiplos saltos para o envio dos dados coletados para um centro de gerência através de uma topologia híbrida. Sendo que a figura do repetidor e do RCH contribuíram diretamente para o aumento da área de cobertura das RSSF's, assim como o número de células possíveis ao longo do *backhaul* de fibra óptica, se comparado com (ASSUMPÇÃO, 2011).

8.2. Propostas de trabalhos futuros

Como propostas para trabalhos futuros é criar uma rede real para realizar a integração entre laboratórios utilizando como *backhaul* a rede Kyatera, e realizar a coleta de grandezas monitoradas em tempo real.

Flexibilização do protocolo de roteamento para busca de melhores rotas de forma automática e não fixa como na proposta implementada.

9 REFERÊNCIAS

1111, Z. Datasheet ZMSW – 1111. 2013. Disponível em: < <http://www.datasheets360.com/part/detail/zmsw-1111/3940557916651725563/> >. Acesso em: maio-2013.

AKYILDIZ, I. F. et al. Wireless sensor networks: a survey. **Computer Networks**, v. 38, n. 4, p. 393-422, 2002. ISSN 1389-1286. Disponível em: < <http://www.sciencedirect.com/science/article/B6VRG-44W46D4-1/2/f18cba34a1b0407e24e97fa7918cdfdc> >.

ANATEL. **REGULAMENTO SOBRE EQUIPAMENTOS DE RADIOCOMUNICAÇÃO DE RADIAÇÃO RESTRITA**. Resolução 506: 33 p. 2008.

ANDRADE, T. P. D. C. D. **Integração de Redes de Sensores sem Fio com tecnologia Rádio-sobre-Fibra**. 2013. 137 (Master Degree). Instituto de Computação, Universidade Estadual de Campinas

ARDUINO. Arduino. 2011. Disponível em: < <http://arduino.cc/> >. Acesso em: 09 Maio 2011.

ASSUMPÇÃO, R. M. D. **Avaliação do impacto em redes de sensores sem fio com utilização de sistema**. 2011. 96 (Master Degree). Engenharia Elétrica, Pontifícia Universidade Católica de Campinas, Campinas.

DING, Z.; YAMAUCHI, N. **An improvement of energy efficient multi-hop time synchronization algorithm in wireless sensor network**. Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference. Beijing, China: 2010: 116-120 p. 2010.

FEDOR, S.; COLLIER, M. **On the Problem of Energy Efficiency of Multi-Hop vs One-Hop Routing in Wireless Sensor Networks**. Advanced Information

Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on. Niagara Falls, Ont. 2: 380-385 p. 2007.

HOSSEN, M. et al. Extension of wireless sensor network by employing RoF-based 4G network. 11th International Conference on Advanced Communication Technology, 2009, 15-18 Feb. p.275-278.

IEEE. IEEE Standard for Information technology- Telecommunications and information exchange between systems- Local and metropolitan area networks- Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). **IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)**, p. 0_1-305, 2006.

INSTRUMENTS, T. **CC1101 Low-Power Sub-1 GHz RF Transceiver (Rev. F)**: Texas INSTRUMENTS: 96 p. 2010.

_____. Design Note DN006. 2013a. Disponível em: < <http://www.ti.com/lit/an/swra123b/swra123b.pdf> >. Acesso em: março-2013.

_____. Smart RF Studio. 2013b. Disponível em: < <http://www.ti.com/tool/SmartRFTM-Studio?keyMatch=smart%20rf%20studio%20cc1101&tisearch=Search-EN-Everything> >. Acesso em: março-2013.

KARL, H.; WILLIG, A. **Protocols and architectures for wireless sensor networks**. Hoboken, NJ: Wiley, 2005. xxv, 497 p. ISBN 0470095105 (cloth alk. paper). Disponível em: < <http://www.loc.gov/catdir/toc/ecip058/2005005800.html> >

KUROSE, J. F.; ROSS, K. W. **Redes de computadores e a internet: uma abordagem top-down**. 3. São Paulo, SP: Pearson Addison Wesley, 2008. 634 p. ISBN 9788588639188.

LATHI, B. P. **Modern digital and analog communication systems**. 3rd. New York: Oxford University Press, 1998. xiii, 781 p. ISBN 0195110099 (cloth). Disponível em: < <http://www.loc.gov/catdir/enhancements/fy0605/97016040-d.html> >

MELLO, A. J. T. S.; LÜDERS, R.; JR., F. N. **Uma Rede de Sensores para Monitoração da Proteção Catódica em Dutos**. 4o PDPETRO. Campinas: ABPG 2007.

MINI-CIRCUITS. **RVA 3000+**.

OIW. Datasheet Antena Omni Vertical 2,4 Ghz. 2013. Disponível em: < <http://www.oiw.com.br/Produto/3/OIW-2408O-SMA.html> >. Acesso em: abril-2013.

RADIUINO. RADIUINO. 2014. Disponível em: < <http://RADIUINO.cc/> >.

RAPPAPORT, T. S. **Wireless communications : principles and practice**. 2nd. Upper Saddle River, N.J.: Prentice Hall PTR, 2002. xxiii, 707 p. ISBN 0130422320. Disponível em: < <http://www.loc.gov/catdir/toc/fy022/2002279109.html> >.

SILVA, D. C. V. **Análise Experimental da Aplicação da Tecnologia de Rádio sobre Fibra em Redes IEEE 802.15.4**. 2009. 108 (Master Degree). Engenharia Elétrica, Pontifícia Universidade Católica de Campinas, Campinas.

ZONU, O. **RF over Fiber Optic Transceiver: OZ810 Series**: Optical Zonu 2010.

ANEXO A – Software de coleta dos testes na bancada.

```
import time

import math

from time import strftime

import serial

from test.test_iterlen import len

#-----

# Classe responsavel pelo teste unitario BLABLABLA

#-----

class SensorTestUnit01:

    #objeto para PER

    per = 0

    #objeto para BER

    ber = 0

    #objeto para Potencia Media DBM RSL

    Pot_med_In_RSI_1 = 0

    #objeto para Potencia Media DBM RBL

    Pot_med_In_RBI_1 = 0

    #objeto para Potencia Media DBM BRL

    Pot_med_In_BRI_1 = 0

    #objeto para Potencia Media DBM SRL

    Pot_med_In_SRI_1 = 0

    def run(self, serialNumberValue, idBase , idSensor, numberOfTest, idFile):

        try:

            # Abrindo a porta serial desejada

            serialPort = serial.Serial(serialNumberValue, 9600, timeout=0.5,parity=serial.PARITY_NONE)

            # Nome do arquivo de log
```



```
filename = strftime(idFile + "_Sensor_%Y_%m_%d_%H-%M-%S.txt")

print "Arquivo de log: %s" % filename

# Abrindo o arquivo de log

fileResource = open(filename, 'w')

# Cria o pacote padrao de envio

packageTest = {}

# Popula o pacote padrao

for index in range(0,52): # Vetor de 52 bytes

    packageTest[index] = 0

    packageTest[8] = int(idSensor)

    packageTest[10] = int(idBase)

    packageTest[16] = 1

    packageTest[19] = 1

    packageTest[22] = 1

    packageTest[25] = 1

# zerando os contadores de numeros de mensagens transmitidas

counterOfSend = 0

# zerando os contadores de numeros de mensagens recebidas completas = 52 bytes

counterOfReceivedPackage = 0

Pot_med_In_RSI_1 = 0

Pot_med_In_RBI_1 = 0

Pot_med_In_BRI_1 = 0

Pot_med_In_SRI_1 = 0

# aguarda 1.5 segundo

time.sleep(1.5)

# limpar buffer serial input

serialPort.flushInput()

# limpar buffer serial output
```

```

serialPort.flushOutput()

FinalBER = 0

# laço principal de envio e recebimento de dados
while counterOfSend < int(numberOfTest):

    #incrementando o numero de mensagens transmitidas
    counterOfSend = counterOfSend + 1

    try:

        index=0

        #Envia pedido para o sensor 1
        for index in range(0,52):

            TXbyte = chr(packageTest[index]) # Deve converter para caracter em ASCII para escrever na serial
            serialPort.write(TXbyte) # Tempo de espera para que receba a resposta do sensor

        #Esta definindo a taxa do PER
        time.sleep(0.3)

        #Recebe pacto enviado acima
        response = serialPort.read(52) # faz a leitura de 52 bytes do buffer que recebe da serial pela COM

        #Verifica se foi recebido o pacote completo, pu seja, 52 bytes
        if len(response) == 52:

            #Incrementa o contador de pacotes recebidos OK
            counterOfReceivedPackage = counterOfReceivedPackage+1

            #processando a resposta
            rssiRS = ord(response[0]) # RSSI_Repetidor x Sensor
            rssiSR = ord(response[1]) # RSSI_Sensor x Repetidor
            rssiRB = ord(response[2]) # RSSI_Repetidor X Base
            rssiBR = ord(response[3]) # RSSI_Base x Repetidor

            #RSSI Repetidor x Sensor byte0
            if rssiRS > 128:

                rssiRS=((rssiRS-256)/2.0)-74

```

```

else:

    rssiRS=(rssiRS/2.0)-74

Pot_med_In_RSI_1 = Pot_med_In_RSI_1 + math.pow(10,(rssiRS/10))

#RSSI Sensor x Repetidor byte1

if rssiSR > 128:

    rssiSR=((rssiSR-256)/2.0)-74

else:

    rssiSR=(rssiSR/2.0)-74

Pot_med_In_SRI_1 = Pot_med_In_SRI_1 + math.pow(10,(rssiSR/10))

#RSSI Retetidor X Base byte2

if rssiRB > 128:

    rssiRB=((rssiRB-256)/2.0)-74

else:

    rssiRB=(rssiRB/2.0)-74

Pot_med_In_RBI_1 = Pot_med_In_RBI_1 + math.pow(10,(rssiRB/10))

#RSSI Base x Repetidor

if rssiBR > 128:

    rssiBR=((rssiBR-256)/2.0)-74

else:

    rssiBR=(rssiBR/2.0)-74

Pot_med_In_BRI_1 = Pot_med_In_BRI_1 + math.pow(10,(rssiBR/10))

print counterOfSend,' S1: RSSI RS =',rssiRS,' RSSI RB =',rssiRB , 'RSSI BR =',rssiBR, 'RSSI SR =',rssiSR

print >>fileResource,time.asctime(),' ',counterOfSend, 'RSSI RS =',rssiRS,' RSSI ReB =',rssiRB , 'RSSI BR
=',rssiBR, 'RSSI SR =',rssiSR

# limpa buffer input serial

serialPort.flushInput()

#aguarda 100 ms

time.sleep(0.1)

if counterOfReceivedPackage == 0:

    #todos as mensagens foram recebidas com erro

    intermediatePER = 0

```

```

else:

    #calcula o PER intermediario para cada transmissao

    intermediatePER = float(1-(float(counterOfReceivedPackage)/float(counterOfSend)))

    print counterOfSend, 'PER_I = ', intermediatePER

    #print >>fileResource,time.asctime(), "      counterOfSend, 'PER_I = ', intermediatePER, '
counterOfReceivedPackage=', counterOfReceivedPackage

except KeyboardInterrupt:

    print 'Abortando Experimento...'

    FinalPER = float(1-(float(counterOfSend)/float(counterOfReceivedPackage)))

    pPot_med_db_RSI_1 = 10* math.log10(Pot_med_In_RSI_1/counterOfReceivedPackage)

    pPot_med_db_RBI_1 = 10* math.log10(Pot_med_In_RBI_1/counterOfReceivedPackage)

    pPot_med_db_BRI_1 = 10* math.log10(Pot_med_In_BRI_1/counterOfReceivedPackage)

    pPot_med_db_SRI_1 = 10* math.log10(Pot_med_In_SRI_1/counterOfReceivedPackage)

        print ' RSSI_media_RS= ', pPot_med_db_RSI_1

    print ' RSSI_media_RB= ', pPot_med_db_RBI_1

    print ' RSSI_media_BR= ', pPot_med_db_BRI_1

    print ' RSSI_media_SR= ', pPot_med_db_SRI_1

    print ' PER= Experimento abortado', FinalPER, ' contador de ciclos= ', counterOfSend, ' contador de pacotes
recebidos= ', counterOfReceivedPackage

    print >>fileResource,time.asctime(),'RSSI_media_RS= ',pPot_med_db_RSI_1

    print >>fileResource,time.asctime(),' RSSI_media_RB= ', pPot_med_db_RBI_1

    print >>fileResource,time.asctime(),' RSSI_media_BR= ', pPot_med_db_BRI_1

    print >>fileResource,time.asctime(),' RSSI_media_SR= ', pPot_med_db_SRI_1

    print >>fileResource,time.asctime(),' PER= Experimento abortado',' contador de ciclos= ', counterOfSend, '
contador de pacotes recebidos= ', counterOfReceivedPackage

    if (FinalPER > 0):

        # Compute BER.

        FinalBER = 1 - ((1-FinalPER) ** 1/512)

        print 'Final BER' , FinalBER

#atualiza atributos

```

```

self.per = FinalPER

self.ber = FinalBER

self.Pot_med_In_RSI_1 = Pot_med_In_RSI_1

self.Pot_med_In_RBI_1 = Pot_med_In_RBI_1

self.Pot_med_In_BRI_1 = Pot_med_In_BRI_1

self.Pot_med_In_SRI_1 = Pot_med_In_SRI_1

#fecha a porta serial

serialPort.close()

#fecha arquivo de log

fileResource.close()

print 'Experimento Aboryado Roteiro de SensorTestUnit1'

break

if counterOfReceivedPackage > 0:

    FinalPER = float(1-(float(counterOfReceivedPackage)/float(counterOfSend)))

    pPot_med_db_RSI_1 = 10* math.log10(Pot_med_In_RSI_1/counterOfReceivedPackage)

    pPot_med_db_RBI_1 = 10* math.log10(Pot_med_In_RBI_1/counterOfReceivedPackage)

    pPot_med_db_BRI_1 = 10* math.log10(Pot_med_In_BRI_1/counterOfReceivedPackage)

    pPot_med_db_SRI_1 = 10* math.log10(Pot_med_In_SRI_1/counterOfReceivedPackage)

    print ' RSSI_media_RS= ', pPot_med_db_RSI_1

    print ' RSSI_media_RB= ', pPot_med_db_RBI_1

    print ' RSSI_media_BR= ', pPot_med_db_BRI_1

    print ' RSSI_media_SR= ', pPot_med_db_SRI_1

    print ' FinalPER= ', FinalPER, ' counterOfSend= ', counterOfSend, ' counterOfReceivedPackage= ',
counterOfReceivedPackage

    print >>fileResource,time.asctime(),'RSSI_media_RS= ',pPot_med_db_RSI_1

    print >>fileResource,time.asctime(),' RSSI_media_RB= ', pPot_med_db_RBI_1

    print >>fileResource,time.asctime(),' RSSI_media_BR= ', pPot_med_db_BRI_1

    print >>fileResource,time.asctime(),' RSSI_media_SR= ', pPot_med_db_SRI_1

    print >>fileResource,time.asctime(),' PER= ', FinalPER, ' counterOfSend= ', counterOfSend, '
counterOfReceivedPackage= ', counterOfReceivedPackage

```

```

if (FinalPER > 0):
    # Compute BER.
    FinalBER = 1 - ((1-FinalPER) ** 1/512)
    print 'Final BER' , FinalBER

serialPort.close()

fileResource.close()

#atualiza atributos

self.per = FinalPER

self.ber = FinalBER

self.Pot_med_In_RSI_1 = Pot_med_In_RSI_1

self.Pot_med_In_RBI_1 = Pot_med_In_RBI_1

self.Pot_med_In_BRI_1 = Pot_med_In_BRI_1

self.Pot_med_In_SRI_1 = Pot_med_In_SRI_1

except Exception,ex:

    print 'Deu bode', ex

# Entrando com a identificacao do Sensor

lenSeries = raw_input("Digite o numero de series de medidas:")

index = 0

# Entrando com o numero da serial

serialNumberPrompt = raw_input("Digite o numero da porta serial:")

# Convertendo o numero da serial em inteiro

serialNumberValue = int(serialNumberPrompt) - 1

# Entrando com a identificacao Base

idBase = raw_input("Digite o ID BASE:")

# Entrando com a identificacao do Sensor

idSensor = raw_input("Digite o ID SENSOR:")

# Entrando com o numero de medidas desejado

numberOfTest = raw_input("Digite o numero de medidas: ")

# Entrando com a identificacao do arquivo de log da sequencia testada

idFile = raw_input("Digite o ID do Arquivo de Log: ")

MedPer = 0

```

```
MedBer = 0

Medpot_med_In_RSI = 0

Medpot_med_In_RBI = 0

Medpot_med_In_BRI = 0

Medpot_med_In_SRI = 0

while index < int(lenSeries):

    # Criando objeto testunitario 1
    serie = SensorTestUnit01()

    # Executando teste unitario1
    serie.run(serialNumberValue, idBase, idSensor, numberOfTest, idFile)

    # soma todas as Ber
    MedBer = MedBer + float(serie.ber)

    # soma todas as Per
    MedPer = MedPer + float(serie.per)

    # soma todas as RSI em Linear
    Medpot_med_In_RSI = Medpot_med_In_RSI + serie.Pot_med_In_RSI_1

    # soma todas as RBI em Linear
    Medpot_med_In_RBI = Medpot_med_In_RBI + serie.Pot_med_In_RBI_1

    # soma todas as BRI em Linear
    Medpot_med_In_BRI = Medpot_med_In_BRI + serie.Pot_med_In_BRI_1

    # soma todas as SRI em Linear
```

```

Medpot_med_In_SRI = Medpot_med_In_SRI + serie.Pot_med_In_SRI_1

index = index + 1

# Nome do arquivo de log
filename = strftime(idFile + "_Sensor_Media_%Y_%m_%d_%H-%M-%S.txt")

print "Arquivo de log: %s" % filename

# Abrindo o arquivo de log
fileResource = open(filename, 'w')

#print resultados finais dos testes BER MEDIO
MedBer = MedBer / int(lenSeries)
print ' BER Medio= ', MedBer
print >>fileResource,time.asctime(),'BER Medio= ',MedBer

#print resultados finais dos testes PER MEDIO
MedPer = MedPer / int(lenSeries)
print ' PER Medio= ', MedPer
print >>fileResource,time.asctime(),'PER Medio= ',MedPer

#print resultados finais dos testes RSI MEDIO
Medpot_med_In_RSI = Medpot_med_In_RSI / int(lenSeries)
pMedpot_med_dBm_RSI = 10* math.log10(Medpot_med_In_RSI)
print ' Medpot_med_dBm_RSI Medio= ', pMedpot_med_dBm_RSI
print >>fileResource,time.asctime(),'Medpot_med_dBm_RSI Medio= ',pMedpot_med_dBm_RSI

#print resultados finais dos testes RBI MEDIO
Medpot_med_In_RBI = Medpot_med_In_RBI / int(lenSeries)
pMedpot_med_dBm_RBI = 10* math.log10(Medpot_med_In_RBI)

```



```

print ' Medpot_med_dBm_RBI Medio= ', pMedpot_med_dBm_RBI

print >>fileResource,time.asctime(),'Medpot_med_dBm_RBI Medio= ',pMedpot_med_dBm_RBI

#print resultados finais dos testes BRI MEDIO

Medpot_med_In_BRI = Medpot_med_In_BRI / int(lenSeries)

pMedpot_med_dBm_BRI = 10* math.log10(Medpot_med_In_BRI)

print ' Medpot_med_dBm_BRI Medio= ', pMedpot_med_dBm_BRI

print >>fileResource,time.asctime(),'Medpot_med_dBm_BRI Medio= ',pMedpot_med_dBm_BRI

#print resultados finais dos testes SRL MEDIO

Medpot_med_In_SRI = Medpot_med_In_SRI / int(lenSeries)

pMedpot_med_dBm_SRI = 10* math.log10(Medpot_med_In_SRI)

print ' Medpot_med_dBm_SRI Medio= ', pMedpot_med_dBm_SRI

print >>fileResource,time.asctime(),'Medpot_med_dBm_SRI Medio= ',pMedpot_med_dBm_SRI

#cerra el archivo de log

fileResource.close()

```

ANEXO B – Firmware do nó sensor B do RCH, camada NET.

```

/**

* Construtor da camada de Rede.

*/

NET::NET()

{

}

/**

* Inicializa a camada de Controle de Acesso ao Meio.

*/

void NET::initialize(void)

{

/*****

```

```

* ENDEREÇO DO SENSOR *
*****/

/* Endereço (ou ID) do sensor, que pode ser escolhido de 0 a 255. */

Net.my_addr = 1;

Net.cel_addr = 1;

/* Caso deseje que o nó receba informações destinadas a qualquer endereço, inicie com o valor 1. */

/* O nó irá receber e tratar todos os pacotes de RF, mesmo que o endereço de destino não seja o dele. */

/* Este parâmetro é útil para a criação de um sniffer de RF, por exemplo. */

Net.disable_addr_check = 0;

/*****

* Fim dos parâmetros RADIUINO *

*****/

}

/**

* Realiza a troca de endereços Origem e Destino

*/

void NET::swapAddresses(packet * pkt)

{

/* Troca os endereços de destino e origem para a retransmissão dos pacotes */

//pkt->NetHdr[0] = pkt->NetHdr[2];

//pkt->NetHdr[1] = pkt->NetHdr[3];

//pkt->NetHdr[2] = Net.my_addr;

//pkt->NetHdr[3] = Net.cel_addr;

}

/**

* Envia o pacote para a camada inferior

*/

void NET::send(packet * pkt)

{

/* Coloca o endereço de origem no pacote. */

//pkt->NetHdr[2] = Net.my_addr;

```

```

//pkt->NetHdr[3] = Net.cel_addr;

pkt->TranspHdr[1] = Net.cel_addr;
/* Envia para a camada inferior */
Mac.send(pkt);
}
/**
 * Recebe o pacote da camada inferior
 */
void NET::receive(packet * pkt)
{
if(pkt->NetHdr[1] == Net.cel_addr || Net.disable_addr_check == 1)
{
/* Envia para a camada superior */
if (pkt->TranspHdr[1] == Net.cel_addr - 1) {
pkt->TranspHdr[1] = Net.cel_addr;
Transp.receive(pkt);
}
}
else if (pkt->NetHdr[1] > Net.cel_addr && pkt->TranspHdr[1] == Net.cel_addr - 1) {
pkt->TranspHdr[1] = Net.cel_addr;
Mac.send(pkt);
}
else if (pkt->NetHdr[1] < Net.cel_addr && pkt->TranspHdr[1] == Net.cel_addr + 1) {
pkt->TranspHdr[1] = Net.cel_addr;
Mac.send(pkt);
}
}
/* Instanciação do objeto de acesso à classe da camada de Rede */
NET Net = NET();

```

ANEXO C – Firmware do nó sensor C do RCH, camada NET.

```

/**
 * Construtor da camada de Rede.
 */
NET::NET()
{
}

/**
 * Inicializa a camada de Controle de Acesso ao Meio.
 */
void NET::initialize(void)
{
    /*****
     * ENDEREÇO DO SENSOR *
     *****/

    /* Endereço (ou ID) do sensor, que pode ser escolhido de 0 a 255. */
    Net.my_addr = 2;
    Net.cel_addr = 0;

    /* Caso deseje que o nó receba informações destinadas a qualquer endereço, inicie com o valor 1. */
    /* O nó irá receber e tratar todos os pacotes de RF, mesmo que o endereço de destino não seja o dele. */
    /* Este parâmetro é útil para a criação de um sniffer de RF, por exemplo. */
    Net.disable_addr_check = 0;

    /*****
     * Fim dos parâmetros RADIUINO *
     *****/
}

```

```
}

/**
 * Realiza a troca de endereços Origem e Destino
 */
void NET::swapAddresses(packet * pkt)
{
    /* Troca os endereços de destino e origem para a retransmissão dos pacotes */
    //pkt->NetHdr[0] = pkt->NetHdr[2];
    //pkt->NetHdr[1] = pkt->NetHdr[3];
    //pkt->NetHdr[2] = Net.my_addr;
    //pkt->NetHdr[3] = Net.cel_addr;
}

/**
 * Envia o pacote para a camada inferior
 */
void NET::send(packet * pkt)
{
    /* Coloca o endereço de origem no pacote. */
    //pkt->NetHdr[2] = Net.my_addr;
    //pkt->NetHdr[3] = Net.cel_addr;

    /* Envia para a camada inferior */
    Mac.send(pkt);
}

/**
 * Recebe o pacote da camada inferior
 */
void NET::receive(packet * pkt)
```

```

{
  if(pkt->NetHdr[3] == 1 || Net.disable_addr_check == 1)
  {
    /* Envia para a camada superior */
    Transp.receive(pkt);
  }
  else
  {
    return;
  }
}

/* Instanciação do objeto de acesso à classe da camada de Rede */
NET Net = NET();

```

ANEXO D – Firmware do nó sensor S, camada NET.

* Construtor da camada de Rede.

```

*/
NET::NET()
{
}
/**
 * Inicializa a camada de Controle de Acesso ao Meio.
 */
void NET::initialize(void)
{
  /******
   * ENDEREÇO DO SENSOR *
   *****/
  /* Endereço (ou ID) do sensor, que pode ser escolhido de 0 a 255. */
  Net.my_addr = 1;
  Net.cel_addr = 1;

```

```

/* Caso deseje que o nó receba informações destinadas a qualquer endereço, inicie com o valor 1. */
/* O nó irá receber e tratar todos os pacotes de RF, mesmo que o endereço de destino não seja o dele. */
/* Este parâmetro é útil para a criação de um sniffer de RF, por exemplo. */
Net.disable_addr_check = 0;

/*****

* Fim dos parâmetros RADIUINO *
*****/

}

/**
* Realiza a troca de endereços Origem e Destino
*/

void NET::swapAddresses(packet * pkt)
{
/* Troca os endereços de destino e origem para a retransmissão dos pacotes */
pkt->NetHdr[0] = pkt->NetHdr[2];
pkt->NetHdr[2] = Net.my_addr;
pkt->NetHdr[1] = pkt->NetHdr[3];
pkt->NetHdr[3] = Net.cel_addr;
}

/**
* Envia o pacote para a camada inferior
*/

void NET::send(packet * pkt)
{
/* Coloca o endereço de origem no pacote. */
pkt->NetHdr[2] = Net.my_addr;
pkt->NetHdr[3] = Net.cel_addr;

/* Envia para a camada inferior */

```

```

    Mac.send(pkt);
}
/**
 * Recebe o pacote da camada inferior
 */
void NET::receive(packet * pkt)
{
    if(pkt->NetHdr[0] == Net.my_addr && pkt->NetHdr[1] == Net.cel_addr || Net.disable_addr_check == 1)
    {
        if (pkt->TranspHdr[1] == Net.cel_addr) {
            pkt->TranspHdr[1] = 0;
            /* Envia para a camada superior */
            Transp.receive(pkt);
        }
    }
    else
    {
        return;
    }
}
/* Instanciação do objeto de acesso à classe da camada de Rede */
NET Net = NET();

```

ANEXO E – Firmware da base, camada NET.

```

/**
 * Construtor da camada de Rede.
 */
NET::NET()
{
}
/**
 * Inicializa a camada de Controle de Acesso ao Meio.

```



```

*/

void NET::initialize(void)

{
    /*****

    * ENDEREÇO DO SENSOR *

    *****/

    /* Endereço (ou ID) do sensor, que pode ser escolhido de 0 a 255. */

    Net.my_addr = 0;

    Net.cel_addr = 0;

    /* Caso deseje que o nó receba informações destinadas a qualquer endereço, inicie com o valor 1. */

    /* O nó irá receber e tratar todos os pacotes de RF, mesmo que o endereço de destino não seja o dele. */

    /* Este parâmetro é útil para a criação de um sniffer de RF, por exemplo. */

    Net.disable_addr_check = 0;

    /*****

    * Fim dos parâmetros RADIUINO *

    *****/

}

/**

* Realiza a troca de endereços Origem e Destino

*/

void NET::swapAddresses(packet * pkt)

{

    /* Troca os endereços de destino e origem para a retransmissão dos pacotes */

    pkt->NetHdr[0] = pkt->NetHdr[2];

    pkt->NetHdr[1] = pkt->NetHdr[3];

    pkt->NetHdr[2] = Net.my_addr;

    pkt->NetHdr[3] = Net.cel_addr;

}

/**

* Envia o pacote para a camada inferior

```

```

*/

void NET::send(packet * pkt)
{
    /* Coloca o endereço de origem no pacote. */
    pkt->NetHdr[2] = Net.my_addr;
    pkt->NetHdr[3] = Net.cel_addr;
    /* Envia para a camada inferior */
    Mac.send(pkt);
}

/**
 * Recebe o pacote da camada inferior
 */

void NET::receive(packet * pkt)
{
    if((pkt->NetHdr[0] == Net.my_addr && pkt->NetHdr[1] == Net.cel_addr) || Net.disable_addr_check == 1)
    {
        if (pkt->TranspHdr[1] == Net.cel_addr + 1) {
            /* Envia para a camada superior */
            Transp.receive(pkt);
        }
    }
}

/* Instanciação do objeto de acesso à classe da camada de Rede */
NET Net = NET();

```

ANEXO F – Software de coleta dos testes externos

```
import math
```

```
import serial

import time

import math

from time import strftime

from datetime import datetime, timedelta

#-----

# Classe principal com as regras de envio e recepcao para todos

# os sensores e celulas configuradas,

# Exemplo : S = Sensor, C = Celula em uma rede de (2)duas celulas com

# (5)cinco sensores por celulas

#

# A cada execucao do metodo run() eh enviado os seguintes comandos

# Comandos TX/RX

# C1S1, C1S2, C1S3, C1S4, C1S5, C2S1, C2S2, C2S3, C2S4, C2S5

#

# Data da ultima alteracao 08.03.2015

#-----

class SensorTestUnit:

    #variavel para PER

    per = 0

    #variavel para BER

    ber = 0

    #variavel para Potencia Media DBM CS

    Pot_med_In_CS = 0

    #variavel para Potencia Media DBM SC

    Pot_med_In_SC = 0

    #variavel para Potencia Media DBM B1B0

    Pot_med_In_B1B0 = 0

    #variavel para Potencia Media DBM B0B1

    Pot_med_In_B0B1 = 0

    #variavel para Potencia Media DBM B1B2
```

```

Pot_med_In_B1B2 = 0

#variavel para Potencia Media DBM B2B1

Pot_med_In_B2B1 = 0

# Metodo principal que executado.....

def run(self, serialNumberValue, qtdCells, qtdSensors, contadorPacotesEnviados, contadorPacotesRecebidos, firstTime):

    serialPort = 0

    try:

        # Abrindo a porta serial desejada

        serialPort = serial.Serial(serialNumberValue, 9600, timeout=0.5, parity=serial.PARITY_NONE)

        # Percorre todas as celulas informadas

        for idCell in range(1,int(qtdCells)+1):

            # Percorre todos os sensores da celulas

            for idSensor in range(1,int(qtdSensors[idCell])+1):

                try:

                    # Nome do arquivo de log

                    filename = strftime(str(idCell) + "_" + str(idSensor) + ".csv")

                    print "Arquivo de log: %s" % filename

                    # Abrindo o arquivo de log

                    fileResource = open(filename, 'a')

                    # Cria o pacote padrao de envio

                    packageTest = {}

                    # Popula o pacote padrao

                    for index in range(0,52): # Vetor de 52 bytes

                        packageTest[index] = 0

                    packageTest[8] = int(idSensor)

                    packageTest[9] = int(idCell)

                    packageTest[10] = 0

                    packageTest[11] = 0

                    Pot_med_In_CS = 0

                    Pot_med_In_SC = 0

```

```

Pot_med_In_B1B0 = 0

Pot_med_In_B0B1 = 0

Pot_med_In_B1B2 = 0

Pot_med_In_B2B1 = 0

Pot_med_In_B2B3 = 0

Pot_med_In_B3B2 = 0

# aguarda 1.5 segundo para escrever serial

time.sleep(1.5)

# limpar buffer serial input

serialPort.flushInput()

# limpar buffer serial output

serialPort.flushOutput()

FinalBER = 0

if firstTime == 0:

    # coloca o cabeçalho no arquivo

    print
    RSSI_CS;','RSSI_SC;','RSSI_B1B0;','RSSI_B0B1;','RSSI_B1B2;','RSSI_B2B1;','PER_I' >>fileResource,'data_hora;','id;',

    contadorPacotesEnviados[int(idCell)-1][int(idSensor)-1] = contadorPacotesEnviados[int(idCell)-
1][int(idSensor)-1] + 1

try:

    #Envia pedido para o sensor 1

    for index in range(0,52):

        TXbyte = chr(packageTest[index]) # Deve converter para caracter em ASCII para escrever na serial

        serialPort.write(TXbyte) # Tempo de espera para que receba a resposta do sensor

    #Espera 1 segundos para ler serial

    time.sleep(1.0)

    #Recebe pacote enviado acima

    response = serialPort.read(52)

```

```

#Verifica se foi recebido o pacote completo

if len(response) == 52:

    #Incrementa o contador de pacotes recebidos OK

    contadorPacotesRecebidos[int(idCell)-1][int(idSensor)-1] = contadorPacotesRecebidos[int(idCell)-
1][int(idSensor)-1] + 1

    #armazena endereco do sensor

    sensor_addr = ord(response[10])

    #armazena endereco do celula

    sensor_cell = ord(response[11])

    #processando a resposta

    rssiCS = ord(response[0]) # RSSI_Repetidor_Celula x Sensor

    rssiSC = ord(response[1]) # RSSI_Sensor x Repetidor_Celula

    rssiB1B0 = ord(response[2]) # RSSI_Repetidor_Backhaul1 X Base

    rssiB0B1 = ord(response[3]) # RSSI_Base x Repetidor_Backhaul1

    rssiB1B2 = ord(response[46]) # RSSI_Repetidor_Backhaul1 x Repetidor_Backhaul2

    rssiB2B1 = ord(response[47]) #RSSI_Repetidor_Backhaul2 x Repetidor_Backhaul1

    rssiB2B3 = ord(response[48]) # RSSI_Repetidor_Backhaul2 x Repetidor_Backhaul3

    rssiB3B2 = ord(response[49]) # RSSI_Repetidor_Backhaul3 x Repetidor_Backhaul2

    #RSSI RSSI_Repetidor_Celula x Sensor byte0

    if rssiCS > 128:

        rssiCS=((rssiCS-256)/2.0)-74

    else:

        rssiCS=(rssiCS/2.0)-74

    Pot_med_In_CS = Pot_med_In_CS + math.pow(10,(rssiCS/10))

    #RSSI RSSI_Sensor x Repetidor_Celula byte1

    if rssiSC > 128:

        rssiSC=((rssiSC-256)/2.0)-74

    else:

```

```

    rssiSC=(rssiSC/2.0)-74

Pot_med_In_SC = Pot_med_In_SC + math.pow(10,(rssiSC/10))

#RSSI_Retetidor_Backhaul1 X Base byte2

if rssiB1B0 > 128:

    rssiB1B0=((rssiB1B0-256)/2.0)-74

else:

    rssiB1B0=(rssiB1B0/2.0)-74

Pot_med_In_B1B0 = Pot_med_In_B1B0 + math.pow(10,(rssiB1B0/10))

#RSSI_Base x Repetidor_Backhaul1 byte3

if rssiB0B1 > 128:

    rssiB0B1=((rssiB0B1-256)/2.0)-74

else:

    rssiB0B1=(rssiB0B1/2.0)-74

Pot_med_In_B0B1 = Pot_med_In_B0B1 + math.pow(10,(rssiB0B1/10))

#RSSI_Repetidor_Backhaul1 x Repetidor_Backhaul2 byte14

if rssiB1B2 > 128:

    rssiB1B2=((rssiB1B2-256)/2.0)-74

else:

    rssiB1B2=(rssiB1B2/2.0)-74

Pot_med_In_B1B2 = Pot_med_In_B1B2 + math.pow(10,(rssiB1B2/10))

#RSSI_Repetidor_Backhaul2 x Repetidor_Backhaul1 byte15

if rssiB2B1 > 128:

    rssiB2B1=((rssiB2B1-256)/2.0)-74

else:

    rssiB2B1=(rssiB2B1/2.0)-74

Pot_med_In_B2B1 = Pot_med_In_B2B1 + math.pow(10,(rssiB2B1/10))

print 'Total Pacotes Enviados' , contadorPacotesEnviados[int(idCell)-1][int(idSensor)-1], ' S1: RSSI CS
=',rssiCS,' RSSI SC =',rssiSC , 'RSSI B1B0 =',rssiB1B0,'RSSI B0B1 =',rssiB0B1,'RSSI B1B2 =',rssiB1B2,'RSSI B2B1
=',rssiB2B1

```

```

        #aguarda 300 ms

        time.sleep(0.3)

        # formata o arquivo

        print >> fileResource, time.asctime() , ' - ID ', sensor_cell, ' / ', sensor_addr, ',',
        contadorPacotesEnviados[idCell-1][idSensor-1], ',', rssiCS, ',', rssiSC, ',', rssiB1B0, ',', rssiB0B1, ',', rssiB1B2, ',', rssiB2B1,

    else:

        print "Erro na resposta do Sensor =" + str(idSensor) + ' / Celula = ' + str(idCell)

        print >> fileResource, time.asctime() , ' - ID - ERRO ', str(idSensor) , ' / ', str(idCell) , ',',
        contadorPacotesEnviados[idCell-1][idSensor-1], ',', 0, ',', 0, ',', 0, ',', 0, ',', 0, ',', 0,

    #calcula o PER intermediario para cada transmissao

    intermediatePER = float(1-(float(contadorPacotesRecebidos[idCell-1][idSensor-1])/float(contadorPacotesEnviados[idCell-1][idSensor-1])))

    print contadorPacotesEnviados[idCell-1][idSensor-1], 'PER_I = ', intermediatePER

    print >> fileResource, ',', intermediatePER

except KeyboardInterrupt:

    print 'Abortando Experimento...'

    #fecha a porta serial

    serialPort.close()

    #fecha arquivo de log

    fileResource.close()

    print 'Experimento Abortado Roteiro de SensorTestUnit1'

    break

except Exception,ex:

    print 'Erro Inesperado!', ex

serialPort.flushInput()

serialPort.flushOutput

serialPort.close()

fileResource.close()

except Exception,ex:

    print 'Erro ao abrir porta serial', ex

    raise

```



```

index = 0

qtdSensors = {}

contadorPacotesEnviados = [[]]

contadorPacotesEnviados = [[]]

# Entrando com a quantidade de series desejada

lenSeries = raw_input("Digite o tempo de testes [minutos]:")

# Entrando com o numero da serial

serialNumberPrompt = raw_input("Digite o numero da porta serial[exemplo: com1 = 1]:")

# Convertendo o numero da serial em inteiro

serialNumberValue = int(serialNumberPrompt) - 1

# Entrando com quantidade de celulas

qtdCells = raw_input("Digite a quantidade de celulas [1,5]:")

# Entrando com quantidade de sensores

for i in range(1,int(qtdCells)+1):

    qtdSensors[i] = raw_input("Digite a quantidade de sensores da celula " + str(i) + " [1,50]:")

# Cria e zera os contadores de envio de pacotes

contadorPacotesEnviados = [[0 for x in range(32)] for x in range(2)]

# Cria e zera os contadores de recebimento de pacotes

contadorPacotesRecebidos = [[0 for x in range(32)] for x in range(2)]

MedPer = 0

MedBer = 0

Medpot_med_In_CS = 0

Medpot_med_In_SC = 0

Medpot_med_In_B1B0 = 0

Medpot_med_In_B0B1 = 0

Medpot_med_In_B1B2 = 0

Medpot_med_In_B2B1 = 0

print 'Iniciando Testes Outdoor ... '

now = datetime.now()

```

```
future = datetime.now();

future = future + timedelta(minutes=int(lenSeries))

while now < future:

    # Criando objeto testunitario

    serie = SensorTestUnit()

    # Executa o teste

    serie.run(serialNumberValue, qtdCells, qtdSensors, contadorPacotesEnviados, contadorPacotesRecebidos, index)

    # soma todas as Per

    MedPer = MedPer + float(serie.per)

    # soma todas as CS em Linear

    Medpot_med_In_CS = Medpot_med_In_CS + serie.Pot_med_In_CS

    # soma todas as SC em Linear

    Medpot_med_In_SC = Medpot_med_In_SC + serie.Pot_med_In_SC

    # soma todas as B1B0 em Linear

    Medpot_med_In_B1B0 = Medpot_med_In_B1B0 + serie.Pot_med_In_B1B0

    # soma todas as B0B1 em Linear

    Medpot_med_In_B0B1 = Medpot_med_In_B0B1 + serie.Pot_med_In_B0B1

    # soma todas as B1B2 em Linear

    Medpot_med_In_B1B2 = Medpot_med_In_B0B1 + serie.Pot_med_In_B1B2

    # soma todas as B2B1 em Linear

    Medpot_med_In_B2B1 = Medpot_med_In_B0B1 + serie.Pot_med_In_B2B1

    index = index + 1

    now = datetime.now()

print 'Testes Finalizados'
```