

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA

MESTRADO PROFISSIONAL EM GESTÃO DE REDES DE
TELECOMUNICAÇÕES

THIAGO TORTORELLI DE FARIA

MULTILINK PARA DETERMINAÇÃO DA TAXA DE
OCUPAÇÃO DE AMBIENTES INTERNOS
UTILIZANDO UMA REDE DE SENSORES SEM FIO

CAMPINAS
2015

THIAGO TORTORELLI DE FARIA

MULTILINK PARA DETERMINAÇÃO DA TAXA DE
OCUPAÇÃO DE AMBIENTES INTERNOS
UTILIZANDO UMA REDE DE SENSORES SEM FIO

Dissertação apresentada como exigência para a
obtenção do título de mestre em Engenharia Elétrica,
ao programa de Pós-Graduação *Strictu Sensu* da
Pontifícia Universidade Católica de Campinas

Orientador: Prof. Dr. Omar Carvalho Branquinho

PUC-CAMPINAS
2015

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

t621.3851
F224m

Faria, Thiago Tortorelli de.

Multilink para determinação da taxa de ocupação de ambientes internos utilizando uma rede de sensores sem fio / Thiago Tortorelli de Faria. - Campinas: PUC-Campinas, 2015.
96p.

Orientador: Omar Carvalho Branquinho.

Dissertação (mestrado) – Pontifícia Universidade Católica de Campinas, Centro de Ciências Exatas, Ambientais e de Tecnologias, Pós-Graduação em Engenharia Elétrica.

Inclui anexo e bibliografia.

1. Redes de sensores sem fio. 2. Sistemas de comunicação sem fio. 3. Telecomunicações. 4. Programação (Computadores). 5. Modelos matemáticos. I. Branquinho, Omar Carvalho. II. Pontifícia Universidade Católica de Campinas. Centro de Ciências Exatas, Ambientais e de Tecnologias. Pós-Graduação em Engenharia Elétrica. III. Título.

22.ed. CDD – t621.3851

THIAGO TORTORELLI DE FARIA

MULTILINK PARA DETERMINAÇÃO DA TAXA DE
OCUPAÇÃO DE AMBIENTES INTERNOS
UTILIZANDO UMA REDE DE SENSORES SEM FIO

Dissertação apresentada como exigência para a
obtenção do título de mestre em Engenharia Elétrica,
ao programa de Pós-Graduação *Strictu Sensu* da
Pontifícia Universidade Católica de Campinas

Orientador: Prof. Dr. Omar Carvalho Branquinho

Dissertação defendida e aprovada em 23 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. Alexandre de Assis Mota
Pontifícia Universidade Católica de Campinas

Prof^a Dr^a Rosa Cristina Cecche Linz
Universidade Estadual de Campinas

Dedico este trabalho ao meu pai
Gilberto, minha mãe Rossana e à minha irmã
Ana Claudia pelo amor, apoio e compreensão
que sempre demonstraram por mim.

AGRADECIMENTOS

Agradeço ao Prof. Dr. Omar Carvalho Branquinho,
Orientador e incentivador no decorrer deste trabalho no Programa de Pós-Graduação em Engenharia Elétrica da Pontifícia Universidade Católica de Campinas, pelo apoio, conselhos, incentivo, atenção e amizade.

Ao Prof. Dr. Alexandre de Assis Mota,
Por todo apoio, opiniões, idéias, sugestões, incentivos e amizade durante momentos mais críticos do trabalho.

Aos colegas mestrandos,
Por todo tempo que passamos juntos, aulas, trabalhos e a troca de conhecimento durante o curso. Em especial à mestranda Karyna Cardoso por todo o companheirismo, incentivo, ajuda e amizade.

Aos professores do curso,
Pela dedicação e transmissão de conhecimento ao longo do curso.

Ao Eduardo Veiga,
Técnico do laboratório de meios de transmissão pela ajuda nos ajustes do setup de testes no laboratório.

Aos alunos de iniciação científica,
Pela ajuda e companheirismo durante alguns testes.

À Pontifícia Universidade Católica de Campinas,
Pela bolsa concedida durante o período do curso de Mestrado Profissional em Gestão de Redes de Telecomunicações.

“No que diz respeito ao empenho, ao compromisso, à dedicação,
não existe meio termo. Ou você faz uma coisa bem feita,
ou não faz.”

Ayrton Senna da Silva

RESUMO

FARIA, Thiago Tortorelli. Multilink Para Determinação da Taxa de Ocupação em Ambientes Internos Utilizando uma Rede de Sensores sem Fio. 2015. Dissertação de Mestrado – Pontifícia Universidade Católica de Campinas, Centro de Ciências Exatas, Ambientais e de Tecnologia, Programa de Pós Graduação em Engenharia Elétrica, Campinas, 2015.

Este trabalho tem por objetivo a obtenção de um modelo matemático para a determinação da taxa de ocupação de ambientes internos baseado na intensidade de sinal de uma Rede de Sensores sem Fio em 915 MHz utilizando quatro links rádio no ambiente. Foi montada uma rede com uma base e quatro nós sensores em um laboratório como intuito de coletar a RSSI (*Received Signal Strength Indication*) de cada link para cada diferente ocupação do espaço para cada grupo de pessoas. Com base nos dados coletados, foram calculados a média, desvio padrão e variância de cada link e cada grupo de pessoas. Esses dados foram utilizados como entrada em modelos matemáticos para a determinação da taxa de ocupação do ambiente. Foram propostos três modelos matemáticos para tal estimação. Os três modelos se mostraram aptos a estimar o número de pessoas utilizando a média geral dos links. Os resultados iniciais mostraram que a tendência é de diminuição da RSSI e o aumento do desvio padrão quanto maior o número de pessoas no ambiente. Alguns links analisados de forma individual se mostraram com uma variação grande e não seguindo inteiramente a tendência mencionada, mas apesar disso a média geral dos links segue essa tendência, ou seja, apesar de um link demonstrar uma grande variação, os outros links tenderam a compensar essa variação. Com a média geral é possível chegar a um erro pequeno entre o número real de pessoas e o número estimado de pessoas. Para a escolha do melhor modelo foi utilizado o RAM (Resíduo Absoluto Médio) e para a média geral dos links o modelo de segunda ordem foi o que se mostrou melhor com um resíduo abaixo de meia pessoa. Por fim os resultados obtidos com *multilink* foram comparados com resultados obtidos em um trabalho em que foi utilizado *single-link* para a obtenção da taxa de ocupação em um ambiente interno. *Multilink*, com um RAM de aproximadamente 0,5 pessoas, se mostrou com um erro menor comparado com o *single-link*, que obteve um erro de aproximadamente duas pessoas.

Palavras-chave: RSSF, Taxa de Ocupação, Ambientes Internos, Modelos Matemáticos, RSSI, Gerência de Redes, MMQ.

ABSTRACT

This work aims to obtain a mathematical model to determinate the occupancy rate in indoor environment based on signal strength of a wireless sensor network (WSN) in 915 MHz using four links at the environment. It was set a network with one sink and four node sensors in a laboratory in order to collect the RSSI (Received Signal Strength Indication) of each individual link for every different space occupation for each group of people. Based on the data collected, it was calculated the mean, standard deviation and the variance of each individual link and the overall average of links for each group of people. The data was used as input in mathematical models to determine the occupancy rate of the environment. Three mathematical models have been proposed to estimate the occupancy rate. All of them proved capable to estimate the number of people using the overall average of links. The results showed that the tendency is to decrease the RSSI and increase the standard deviation the greater the number of people in the environment. Some links analyzed individually showed a large variation and not following entirely the tendency mentioned. Nevertheless the overall average of links follows this tendency showing that despite a link shows a great variation, the other links tended to compensate this variation and with the overall average of links is possible to get a small error between the real number of people and the estimated number of people. To choose the best model it was used the MAE (Mean Absolute Error). The second order model was the best model with a MAE slightly below of a half people. The results obtained using multilink were compared with a work that used single-link to predict the number of people in a indoor environment. Multilink had a smaller error compared with single-link, which obtained an error of about two people.

Keyword: WSN, wireless sensor network, occupancy rate, indoor environment, mathematical models, RSSI, Network Management.

LISTA DE FIGURAS

Figura 1 - Taxa de ocupação de um terreno - 40% do terreno ocupado.	18
Figura 2 - Layout vista de cima do ambiente de testes.....	31
Figura 3 - Laboratório onde foram realizados os experimentos.....	31
Figura 4 - Modulo UART	32
Figura 5 - BE900	32
Figura 6 - Antena monopolo vertical	33
Figura 7 - Placa do kit DK104.....	33
Figura 8 - Analisador de Espectro utilizado na calibração dos sensores.....	35
Figura 9 - Etapa de aferição do medidor de potência do nó sensor	36
Figura 10 - Última etapa aferição medidor de potência do nó sensor	36
Figura 11 - Fluxograma de funcionamento do script de coleta de dados python	37
Figura 12 - Protocolo de Roteamento fixo 4 sensores.	38
Figura 13 - Fluxograma do protocolo de roteamento.....	40
Figura 14 - Gráfico RSSI Média, Desvio Padrão e Variância normalizados do link Base – Sensor 1	44
Figura 15 - Comparação entre número real de pessoas e número estimado de pessoas pelo modelo de 1° ordem	47
Figura 16 - Gráfico RSSI Média, Desvio Padrão e Variância normalizados do link Sensor 1 – Sensor 2.....	48
Figura 17 - Comparação entre número real de pessoas e número estimado pelo modelo de 2° ordem - Link 12	51
Figura 18 - Gráfico RSSI Média, Desvio Padrão e Variância normalizados do link Sensor 2 – Sensor 3.....	52
Figura 19 - Comparação entre número real de pessoas e número estimado de pessoas para o link 23.....	55
Figura 20 - Gráfico RSSI Média, Desvio Padrão e Variância normalizados do link Sensor 3 – Sensor 4.....	56
Figura 21 - Comparação entre número real de pessoas e o número estimado de pessoas utilizando o modelo de 2° ordem para o link sensor 3-sensor 4.	59
Figura 22 - Média, Desvio Padrão e Variância - Média Geral.....	60
Figura 23 - Comparação entre número real de pessoas e o número estimado de pessoas utilizando o modelo de 2° ordem – Média geral dos links.	63
Figura 24 - Vista de cima do ambiente de testes efetuados por (BRAGA, DANIEL B.) - Disposição do link.....	66
Figura 25 - Comparação do resíduo absoluto médio encontrado em (BRAGA, DANIEL B.) com single-link e encontrado neste trabalho com multilink.	68

LISTA DE TABELAS

Tabela 1 - Média, Desvio Padrão, Variância e Média Quadrática normalizados do Link Base – Sensor 1.....	44
Tabela 2 - Parâmetros estimados para os modelos do link: Base - Sensor 1.....	45
Tabela 3 - Número estimado de pessoas em cada modelo para o link: Base - Sensor 1.....	45
Tabela 4 - Resíduo absoluto médio de cada modelo do Link: Base - Sensor 1.....	46
Tabela 5 - Resíduo absoluto com o modelo de 1º ordem do link: Base- Sensor 1.....	46
Tabela 6 - Média, Desvio Padrão, Variância e Média Quadrática normalizados do link Sensor 1 - Sensor 2.....	48
Tabela 7 - Parâmetros estimados para o link: Sensor 1 - Sensor 2.....	49
Tabela 8 - Número estimado de pessoas de cada modelo para o link: Sensor 1 - Sensor 2.....	49
Tabela 9 - Resíduo absoluto médio dos três modelos para o link: Sensor 1 - Sensor 2.....	49
Tabela 10 - Resíduo absoluto encontrado utilizando o modelo de 2º ordem para o link: Sensor 1 - Sensor 2.....	50
Tabela 11 - Média, Desvio Padrão, Variância e Média Quadrática normalizados do link Sensor 2 - Sensor 3.....	52
Tabela 12 - Parâmetros estimados para cada um dos modelos propostos – Link Sensor 2–Sensor 3.	53
Tabela 13 - Número estimado de pessoas para cada modelo - Link: Sensor 2 - Sensor 3.....	53
Tabela 14 - Resíduo absoluto médio dos três modelos - Link: Sensor 2 - Sensor 3.....	54
Tabela 15 - Resíduo absoluto encontrado para cada grupo de pessoas.	54
Tabela 16 - Média, Desvio Padrão, Variância e Média Quadrática normalizados do link Sensor 3 - Sensor 4.....	56
Tabela 17 – Parâmetros estimados para cada um dos modelos propostos – Link Sensor 3–Sensor 4.....	57
Tabela 18 - Número estimado de pessoas para cada modelo - Link: Sensor 3 - Sensor 4.....	57
Tabela 19 - Resíduo absoluto médio para cada modelo - Link: Sensor 3 - Sensor 4.	58
Tabela 20 - Resíduo absoluto - Link: Sensor 3 - Sensor 4.....	58
Tabela 21 - Valores de entrada calculados - Média geral dos links.	60
Tabela 22 - Parâmetros estimados para cada um dos modelos propostos – Média Geral.	61
Tabela 23 - Número estimado de pessoas em cada modelo – Média Geral.	62
Tabela 24 - Resíduo Absoluto Médio para cada modelo – Média Geral.	62
Tabela 25 - Resíduo Absoluto para o modelo de 2º ordem - Média Geral dos links.	62
Tabela 26 - Número estimado de pessoas utilizando <i>single-link</i> em BRAGA (2014).	67
Tabela 27 - Resíduo absoluto médio encontrado em cada modelo proposto em BRAGA (2014)...	67

LISTA DE ABREVIATURAS E SIGLAS

RSSF	= Redes de Sensores sem Fio
RSSI	= <i>Received Signal Strength Indication</i>
MMQ	= Método dos Mínimos Quadrados
RS	= Resíduo Simples
RM	= Resíduo Médio
RAM	= Resíduo Absoluto Médio
RQM	= Resíduo Quadrático Médio
MHz	= Mega Hertz
dB	= Decibel
dBm	= Decibel em mili-watt
FSK	= <i>Frequency Shift Keying</i>
mW	= milli-watt
USB	= <i>Universal Serial Buss</i>
RF	= Rádio Frequência
ISM	= <i>Industrial, Scientific and Medical</i>
Link B1	= Link Base – Sensor 1
Link 12	= Link Sensor 1 – Sensor 2
Link 23	= Link Sensor 2 – Sensor 3
Link 34	= Link Sensor 3 – Sensor 4

SUMÁRIO

1. INTRODUÇÃO.....	14
1.1. Contextualização do Problema.....	14
1.2. Justificativa.....	15
1.3. Objetivo.....	15
1.4. Organização do Trabalho.....	15
2. REVISÃO BIBLIOGRÁFICA.....	17
2.1. Redes de Sensores sem fio e suas aplicações.....	17
2.2. Taxa de ocupação.....	18
2.3. Propagação de sinais em ambientes fechados (<i>indoor</i>).....	21
2.4. Outros fenômenos que influenciam RSSF em um ambiente fechado.....	21
3. PROPOSTA DO TRABALHO.....	23
4. METODOLOGIA.....	30
4.1. Ambiente de Testes e Equipamentos Utilizados.....	30
4.2. Calibração dos rádios.....	34
4.3. Processo de coleta de dados.....	36
5. RESULTADOS.....	42
6. COMPARAÇÃO ENTRE <i>MULTILINK</i> E <i>SINGLE-LINK</i>	65
7. CONCLUSÃO.....	70
REFERÊNCIAS BIBLIOGRÁFICAS.....	72
ANEXO A – Software de coleta de Dados Python.....	74
ANEXO B – Protocolo de Roteamento Fixo de 4 saltos.....	80
ANEXO C – Código de tratamento de dados no Scilab.....	88
ANEXO D – Mapa do pacote Radiuino – 52 Bytes.....	96

1. INTRODUÇÃO

As RSSF (Redes de Sensores sem Fio) estão se tornando uma realidade nos dias de hoje. Sua função é medir grandezas e controlar processos. Entretanto novas aplicações serão possíveis, na medida do crescimento da utilização da “internet das coisas” para ajudar em problemas do cotidiano.

Um exemplo é a avaliação da taxa de ocupação de ambientes internos em que não se tem um modelo eficaz para esta avaliação. O uso de RSSF pode ser uma alternativa para tal avaliação. Em capítulo posterior será introduzido RSSF e suas aplicações, conceito sobre taxa de ocupação e exemplos de estimação de pessoas em ambientes abertos e fechados.

Neste capítulo será apresentada a contextualização do problema, a justificativa do trabalho, o objetivo e a organização do trabalho.

1.1. Contextualização do Problema

Com o acontecimento de grandes eventos, tanto em ambientes abertos como em ambientes fechados, a estimação do número de pessoas se faz cada vez mais necessária.

Um exemplo é o protesto que aconteceu no dia 15 de Março de 2015 em diversas capitais do Brasil. Segundo SITE UOL (2015), a polícia militar estimou em 1 milhão de pessoas. Já o instituto de pesquisa Datafolha estimou que por volta de 200 mil pessoas estiveram na Avenida Paulista em São Paulo neste dia.

Possivelmente os diferentes métodos de contagem geram essa discrepância, ou talvez interesses políticos possam contribuir para essa diferença.

Em ambientes fechados, um local de grande interesse é na área médica em que, em um pronto socorro ou sala de cirurgia, o número de pessoas no ambiente pode ser muito grande, podendo haver também alta rotatividade de pessoas (STROPARO et al, 2004). Diversos outros tipos de ambientes fechados também são de grande interesse.

No Item 2.2 deste trabalho serão dados alguns exemplos de taxa de ocupação em ambientes internos.

1.2. Justificativa

Dado o problema da contagem do número de pessoas em ambientes abertos e ambientes fechados, a utilização da tecnologia pode ser de grande utilidade uma vez que são escassas as técnicas de estimação de pessoas em tempo real.

Com uma RSSF é possível fazer esta estimativa como mostra BRAGA (2014) que utilizou um simples link de rádio de uma RSSF para determinar a taxa de ocupação de um ambiente interno, em que o erro encontrado entre o número estimado de pessoas e o número real de pessoas foi de 2 pessoas em média.

Neste trabalho é proposto um avanço dos modelos avaliados em BRAGA (2014) uma vez que agora será utilizado *multilink* para a análise e predição do número de pessoas. Também será considerado além da RSSI (*Received Signal Strength Indication*) média e a sua variância, o desvio padrão da RSSI como entrada em um dos modelos. Além dos modelos de 1° e 2° ordem, um modelo híbrido de 3° ordem será proposto para fazer a estimação do número de pessoas.

1.3. Objetivo

Este trabalho tem por objetivo obter um modelo matemático que visa estimar o número de pessoas em um ambiente fechado utilizando *multilink* com uma RSSF. Diversos nós sensores serão espalhados em posições estratégicas no ambiente em questão. Será feita uma comparação entre os resultados obtidos com a utilização do *multilink* e resultados obtidos em BRAGA (2015) utilizando *single-link* com o objetivo de verificar se com a utilização do *multilink* é obtido um ganho para a estimação do número de pessoas.

1.4. Organização do Trabalho

No Capítulo 2 é feita uma revisão bibliográfica sobre Redes de sensores sem fio e suas aplicações, conceitos sobre taxa de ocupação e métodos para

estimar número de pessoas em eventos abertos e ambientes fechados e uma breve explicação sobre propagação de sinais em ambientes fechados.

No Capítulo 3 é apresentada a proposta do trabalho e o conceito sobre modelagem matemática, utilizada na proposta.

No Capítulo 4 é apresentada a metodologia utilizada nos testes para coleta de dados que contempla os ambientes em que os testes foram feitos, equipamentos utilizados, calibração dos sensores, o processo da coleta de dados e o firmware utilizado na base e nos nós sensores.

Todos os resultados obtidos nos testes e suas análises são apresentados no Capítulo 5.

No Capítulo 6 é feita uma comparação de resultados entre estimação de pessoas utilizando *single-link* e *multilink*.

Por fim, no Capítulo 7 é apresentada a conclusão deste trabalho.

2. REVISÃO BIBLIOGRÁFICA

Neste capítulo primeiramente será feita uma breve introdução sobre redes de sensores sem fio e sua aplicação.

O item a seguir fala sobre o conceito de taxa de ocupação e métodos para estimar número de pessoas em eventos em ambientes abertos e ambientes fechados.

Posteriormente será apresentada uma breve introdução sobre propagação de sinais em ambientes fechados.

No final do capítulo são introduzidos outros fenômenos que podem influenciar a RSSI em um ambiente fechado.

2.1. Redes de Sensores sem fio e suas aplicações

Uma necessidade nos dias atuais é a de se ter o controle sobre os processos, monitorar ambientes e outras ações relacionadas ao ambiente físico que possuímos. Muitas vezes necessitamos ter este tipo de monitoramento a grandes distâncias e o uso de fios se torna um grande obstáculo (HOLGER et al, 2005).

Redes de Sensores sem Fio (RSSF) são capazes de atuar em diferentes aplicações do dia a dia, como detecção de movimento, controle de estoque, temperatura de ambiente, etc. Uma RSSF consiste em nós individuais que se comunicam entre si, interagindo com o ambiente em que estão, seja controlando um parâmetro físico, seja por sensibilidade do ambiente (Ilyas et al, 2004).

Alguns exemplos de aplicações são: controlar umidade nos campos de agricultura, monitorando máquinas para efetuar manutenção preventiva ao invés de manutenção corretiva, controlar sistemas prediais como ar condicionado, etc (HOLGER et al, 2005).

As RSSF são empregadas em diversos tipos diferentes de ambiente, tanto em ambientes abertos, ou exteriores (*outdoor*), como na área agrícola, também em ambientes fechados, ou interiores (*Indoor*), como na área médica, área industrial, etc.

Além das aplicações citadas anteriormente, RSSF podem ser utilizadas para entender características de um ambiente. Taxa de ocupação é uma possível característica de um ambiente em que uma RSSF pode ser útil.

O próximo tópico trará o conceito de taxa de ocupação e alguns métodos existentes para sua estimação.

2.2. Taxa de ocupação

O conceito de taxa de ocupação é amplamente utilizado na construção civil e pode ser definido como a porcentagem da área de um terreno em que há edificação, ou seja, é a área total construída do pavimento térreo dividida pela área total do terreno (ABUSSANDRA, 2007). Segue um exemplo de taxa de ocupação na Figura 1.

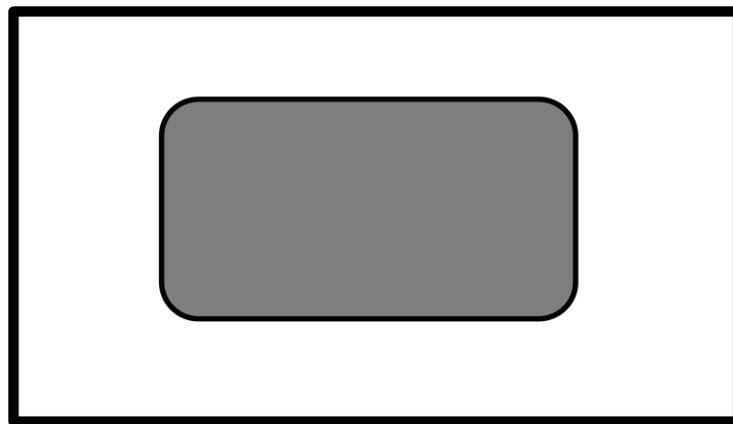


Figura 1 - Taxa de ocupação de um terreno - 40% do terreno ocupado.

Na Figura 1 o quadrado maior (vazado) é a área total de um terreno qualquer e a área sombreada é o total da área construída, que equivale a 40% do total do terreno. Esta ilustração foi feita somente para auxiliar no conceito de taxa de ocupação.

O cálculo da taxa de ocupação é de grande importância em um projeto de construção civil e é feita no início de um projeto. Este tipo de taxa de ocupação é considerada estática.

Um exemplo da utilização do conceito de taxa de ocupação acontece na área médica. Segundo STROPARO et al (2004) é feito um estudo sobre a taxa de ocupação em um centro cirúrgico e é proposto um método para modelar um ambiente cirúrgico com o objetivo de melhorar a taxa de ocupação do mesmo (STROPARO et al 2004).

Este tipo de taxa de ocupação é considerado dinâmico uma vez que há alteração no número de pessoas em uma determinada sala ou centro cirúrgico e também devido à alta movimentação. A definição sobre taxa de ocupação dinâmica pode ser vista na Equação 1.

$$TO_{Din} = \frac{N_{pessoas}}{M^2} \quad (1)$$

Onde: TO_{Din} é a taxa de ocupação dinâmica em um ambiente interno, $N_{pessoas}$ é o número de pessoas e M^2 é metro quadrado do ambiente.

2.2.1. Métodos para estimar número de pessoas em grandes eventos em ambientes abertos

Devido ao grande número de eventos existentes, sejam comícios, shows ou protestos, a estimativa do número de pessoas se mostra importante. Existem vários métodos para se fazer tal estimativa. Um método que é muito utilizado ainda nos dias de hoje é o método de Jacobs, criado por Herbert Jacobs. Este método, que foi aperfeiçoado anos depois por Seidler, Meyer e Gillivray em 1976, utiliza-se de 6 passos (Swank e Clapp, 1999), como segue:

1. Observar a fotografia da multidão de um determinado ponto.
2. Aplicar grades simétricas em toda a região em que a multidão se encontra.
3. Contar (ou estimar) o número de pessoas em cada grade.
4. Verificar o número de grades em toda a região em que se encontra a multidão.
5. Multiplicar o número de pessoas em cada grade pelo número total de grades estimadas.

6. O resultado obtido é o número estimado de pessoas.

2.2.2. Estimação de pessoas em ambientes fechados

Em ginásios, cinemas, em um centro cirúrgico de um hospital e outros, é comumente mais fácil de contar o número de pessoas que estão nestes ambientes, sejam por controle de senhas, ou por catracas de controle na entrada dos ambientes, dentre outros.

Porém nem sempre é possível ter este controle, seja por causa da ausência da utilização destas maneiras de contagem ou pelo alto número de pessoas que estão no evento em que os organizadores não conseguem ter o controle efetivo do número de pessoas presentes.

Em casos como o descrito acima, o uso da tecnologia sem fio para fazer tal contagem pode ser um método a ser implementado, utilizado em tempo real.

Já existem casos em que a tecnologia é utilizada para detectar presença de pessoas e também fazer a contagem delas. Seguem alguns exemplos:

De acordo com SUBBURAMAN et al (2012) que propõe um método que se chama “*head detector*” e utiliza um algoritmo que, a partir de uma imagem recém capturada, detecta a cabeça de cada pessoa presente no local e faz a contagem do número de pessoas . É claro que, neste cenário, não é possível ter a contagem do número de pessoas em tempo real pela necessidade de uma foto do local, mas este é somente um exemplo de uso de tecnologia para contar pessoas em um ambiente fechado.

Outro exemplo do uso da tecnologia visando contar número de pessoas em ambiente interno está em (AKHLAGHINIA, 2009) em que é utilizada uma RSSF e infravermelho para detectar a presença de uma pessoa dentro de um ambiente. O infravermelho é utilizado na presença da pessoa no ambiente e o nó sensor é utilizado na detecção, pelo contato da porta, da entrada de pessoas no ambiente.

2.3. Propagação de sinais em ambientes fechados (*indoor*)

Este item tem por objetivo apresentar um breve conceito sobre propagação de ondas eletromagnéticas em ambientes fechados. Não será apresentado em detalhes por não ser o objetivo final deste trabalho.

A propagação de sinais em ambientes fechados se caracteriza por curtas distâncias, se comparadas com propagação em ambientes abertos (considerando telefonia móvel, por exemplo), e a variação do sinal é maior que em ambientes abertos (Rappaport, 2004). Essa variação se dá por conta dos objetos que se encontram nos ambientes, dentre outros aspectos.

Alguns fenômenos físicos podem atuar na característica de um sinal propagado em um ambiente fechado. Os principais fenômenos são: reflexão, difração e dispersão (Rappaport, 2004). Estes fenômenos podem gerar atenuação do sinal transmitido.

A reflexão acontece quando uma onda eletromagnética propagada em um dado ambiente se choca com um obstáculo que é muito maior que o comprimento de onda da dada onda propagada. Essa onda reflete no obstáculo.

A difração se dá quando existe entre o transmissor e o receptor um objeto em que na superfície deste objeto existam orifícios irregulares e obstruções, fazendo com que o sinal, ao colidir com o objeto (obstáculo) acompanhe a curvatura da obstrução.

A dispersão ocorre quando a obstrução entre o transmissor e o receptor é menor que o comprimento de onda da onda propagada. Quando a onda (o sinal transmitido) colide no obstáculo, ela muda de direção.

2.4. Outros fenômenos que influenciam RSSF em um ambiente fechado

Além dos fenômenos físicos apresentados no Item 2.3, a característica do ambiente e o fluxo de pessoas também podem influenciar no sinal propagado de uma RSSF.

A característica do ambiente se dá pelos objetos que estão no ambiente em que se encontra a RSSF e também pela disposição dos mesmos. Além disso,

o sinal pode ter uma variação se a disposição dos objetos for mudada fazendo com que, por exemplo, a linha de visada em determinado link da rede seja obstruído por um objeto.

O fluxo de pessoas pode influenciar se a movimentação delas dentro de uma sala em que tenhamos uma RSSF for muito alta. Por exemplo, suponha que em uma sala existem 25 pessoas. Essas pessoas estão, em um primeiro momento, estáticas na sala, sem se movimentar. Depois de um tempo as mesmas 25 pessoas começam a se movimentar dentro desta mesma sala. Comparando os dois cenários, é possível que no segundo cenário, em que as pessoas estão se movimentando, o sinal transmitido pela rede terá uma variação maior comparando-se com o primeiro cenário.

3. PROPOSTA DO TRABALHO

Modelos matemáticos podem ser muito úteis para representar matematicamente um sistema real. Um modelo pode representar algumas das características de um sistema real, mas nem todas essas características podem ser representadas. Muitas vezes, para poder representar um sistema real é necessário o uso de mais de um modelo matemático (MOTA, 2005).

A modelagem de um sistema real pode ser importante para que se possa ter um melhor conhecimento do sistema a ser modelado (COTA, 2010).

Como já mencionado em capítulos anteriores, uma necessidade atual é a determinação do número de pessoas em um determinado evento, protesto ou até mesmo em ambientes fechados em que o uso de controle de pessoas não é adequado ou não é utilizado.

A modelagem matemática pode ser muito útil neste cenário aliando-se com a tecnologia.

A proposta deste trabalho é, utilizando-se de RSSF em um ambiente fechado, poder encontrar um modelo matemático que consiga prever o número de pessoas no ambiente em questão a partir da RSSI e da variância do mesmo.

Neste capítulo será feita uma introdução sobre modelos matemáticos, as etapas para a construção de um modelo matemático e por último a modelagem proposta neste trabalho.

3.1. Modelos Matemáticos

Os Itens 3.1.1 a 3.1.5 abordam alguns tipos de modelagem existente e será feita uma breve explicação retirada de MOTA (2005). O Item 3.1.6 fala sobre qual modelagem matemática foi escolhida para utilização neste trabalho. O Item 3.1.7 traz a forma de identificar sistemas. O método dos mínimos quadrados é introduzido no Item 3.1.8. E o Item 3.1.9 discorre sobre a modelagem proposta neste trabalho.

3.1.1. Modelagem Caixa Branca

A modelagem caixa branca também pode ser interpretada como modelagem fenomenológica (Mota, 2005). Para poder se utilizar desta modelagem é necessário que o modelador tenha profundo conhecimento de como o sistema a ser modelado se comporta. Por ser um tipo de modelagem difícil e que necessite de muito conhecimento sobre o sistema modelado, nem sempre se deve empregar este tipo de modelagem.

3.1.2. Modelagem Caixa Preta

Na modelagem caixa preta, ou modelagem empírica, não existe a necessidade de se ter conhecimento sobre o sistema a ser modelado. Ele parte do princípio em que existe um banco de dados com dados reais medidos no sistema a ser modelado.

É um tipo de modelagem em que os modelos obtidos são mais simples e mais fáceis de utilizar, porém existem algumas desvantagens como a presença de ruídos no sistema em que se obtiveram os dados e utilização de forma errônea dos dados reais do sistema (banco de dados).

3.1.3. Modelagem Caixa Cinza

A modelagem caixa cinza é um tipo que se encontra entre as modelagens caixa branca e caixa preta onde se conhecem algumas características do sistema real. São utilizadas algumas equações matemáticas e também se utiliza de um banco de dados para se chegar a um modelo.

3.1.4. Abordagem *Top-Down*

Neste tipo de modelagem, o sistema a ser modelado não é conhecido. Seus componentes não são conhecidos e a visão é de cima para baixo. Vê o sistema como um todo. Ele modela primeiro o sistema como um todo partindo depois para a modelagem de seus componentes.

3.1.5. Abordagem *Bottom-Up*

Já a abordagem *Bottom-Up*, de forma contrária à abordagem anterior, modela o sistema a partir de seus componentes que compõem o sistema, modelando por último o sistema como um todo.

Esta abordagem necessita de conhecimento das partes do sistema por parte do modelador.

3.1.6. Forma de modelagem escolhida

Neste trabalho foi escolhida a utilização do tipo de modelagem caixa preta agregada à abordagem *top-down*, ou seja, o sistema real será modelado a partir de uma base de dados obtida em testes reais para a coleta dos dados necessários.

Esta escolha se deu pelo fato do não conhecimento do comportamento do sistema real a ser modelado, no caso, o laboratório em que os testes foram efetuados.

3.1.7. Identificação de Sistemas

Identificação de sistemas é o nome dado para estimar um modelo matemático caixa preta.

Ela pode ser dividida em 5 etapas (AGUIRRE et al, 1998) :

- Coleta de dados;
- Escolha do tipo de modelo;
- Escolha da estrutura do modelo;
- Estimação de parâmetros;
- Validação do modelo.

A coleta de dados, primeira etapa, acontece no ambiente (sistema real) em que se quer estimar um modelo. Nesta etapa é muito importante a observação por parte do modelador, pois os dados devem ser adquiridos em condições normais do sistema, tomando cuidado com possíveis variáveis externas (PROCÓPIO, 2012). Para exemplificar, no caso do laboratório em que os testes foram efetuados, as condições normais dele seria estar com todas as mesas dispostas nos lugares em que geralmente ficam e não mudar sua configuração habitual.

A segunda etapa é escolher o tipo de modelo. Nesta etapa, de acordo com as características dos dados coletados, é escolhido um modelo inicial. Esta escolha pode ser feita plotando um gráfico com os dados da base de dados do sistema a ser modelado.

Já na terceira etapa é feita a escolha da estrutura do modelo. Escolher a estrutura do modelo significa verificar qual a melhor ordem do modelo matemático que se aplica ao comportamento dos dados do sistema real que foram coletados.

A quarta etapa visa estimar os parâmetros de entrada que representem o comportamento do sistema real. Para esta etapa diversos métodos podem ser utilizados, como o MMQ (Método dos mínimos quadrados), Filtro de Kalman (Mota, 2005).

Na quinta etapa é feita a validação dos modelos de acordo com os parâmetros calculados na etapa anterior.

3.1.8. Método dos Mínimos Quadrados

O método dos mínimos quadrados é um método que tem por objetivo a estimação de parâmetros de saída tendo como base dados experimentais, segundo BRANDOLT (2002).

Com a Equação 2 é possível representar um dado sistema com entradas e saídas.

$$Y = X * \theta \quad (2)$$

Onde:

Y: vetor de saídas, possui n linhas.

X: vetor de entradas, possui n linhas.

θ : vetor dos parâmetros para serem estimados.

Os vetores X e Y, na Equação 1 são obtidos a partir de um sistema real. É interessante introduzir o conceito de resíduo.

O resíduo simples, o mais comum entre eles, é um vetor de erros entre os dados reais do sistema e os dados estimados pelo modelo em questão. Além do resíduo simples, existem mais três tipos de resíduos, são eles:

Resíduo simples:
$$\begin{bmatrix} RS_1 \\ RS_2 \\ RS_N \end{bmatrix} = \begin{bmatrix} Y_{medido\ 1} \\ Y_{medido\ 2} \\ Y_{medidoN} \end{bmatrix} - \begin{bmatrix} Y_{estimado\ 1} \\ Y_{estimado\ 2} \\ Y_{estimadoN} \end{bmatrix} \quad (3)$$

Resíduo médio:
$$RM = \frac{1}{N} \sum_{i=1}^N RS_i \quad (4)$$

Resíduo absoluto médio:
$$RAM = \frac{1}{N} * \sum_{i=1}^N |RS_i| \quad (5)$$

Resíduo quadrático médio:
$$RQM = \frac{1}{N} \sum_{i=1}^N RS_i^2 \quad (6)$$

A modelagem matemática visa a obtenção de um modelo que mais possa se ajustar ao comportamento de um sistema real de forma a representar este comportamento com o menor resíduo possível.

Dentre os resíduos apresentados, qualquer um deles pode ser utilizado na validação de um modelo e deve ser escolhido o que melhor se ajustar aos dados. No caso deste trabalho o Resíduo absoluto médio (RAM) é o que será utilizado na validação e comparação dos modelos.

Este resíduo é igual à média dos valores absolutos da diferença entre os valores reais e valores estimados.

A utilização deste resíduo para a validação se justifica, pois neste tipo de análise os dados são tratados todos na forma positiva, valor absoluto, ou seja, o número estimado de pessoas pode ser maior ou menor que o número real de pessoas, dado isto, com o RAM, em ambos os casos o resíduo retornado será positivo e, portanto, este resíduo se mostra o mais adequado.

Outra importante fase na utilização do MMQ para estimação dos parâmetros é o cálculo do vetor de parâmetros (Θ).

A Equação 7 indica como é feita a estimação do vetor de parâmetros.

$$\theta_{\text{estimado}} = [X^T * X]^{-1} * X^T * Y \quad (7)$$

3.1.9. Modelagem Proposta

A proposta deste trabalho é encontrar um modelo matemático a partir de medições de RSSI em um ambiente interno utilizando 4 links de uma rede de sensores sem fio em que seja possível prever o número de pessoas no ambiente. Os dados de entrada utilizados, a partir da RSSI, serão a média da RSSI, desvio padrão e variância. Primeiramente será feita uma análise link a link com os dados de entrada e por último a média geral da RSSI de todos os links será utilizada como entrada nos modelos.

Como já citado anteriormente, em BRAGA (2014) foi proposta a determinação da taxa de ocupação de ambientes internos a partir da RSSI média e variância, utilizando somente um link no ambiente. Como resultado, o modelo

linear de segunda ordem obtido em BRAGA (2014) foi o que melhor se mostrou com um resíduo um pouco menor que 2, ou seja, um erro de aproximadamente 2 pessoas, em média.

Em BRAGA (2014), o modelo linear de 3° ordem se mostrou ineficaz.

Portanto, neste trabalho serão utilizados os modelos lineares de 1° e 2° ordem além de um terceiro modelo híbrido, que utiliza como entrada a RSSI média e a variância.

Os três modelos são apresentados nas Equações (8), (9) e (10).

$$1^\circ \text{Ordem: } Y_{e1} = X_1 * \theta_{\text{média}} + (X_2)^2 * \theta_{\text{var}} + \theta_{\text{constante}} \quad (8)$$

$$2^\circ \text{Ordem: } Y_{e2} = X_1 * \theta_{\text{média}} + (X_1)^2 * \theta_{\text{média } 2} + X_2 * \theta_{\text{desvp}} + (X_2)^2 * \theta_{\text{var}} + \theta_{\text{constante}} \quad (9)$$

$$3^\circ \text{Ordem: } Y_{e3} = X_1 * \theta_{\text{média}} + (X_1)^2 * \theta_{\text{média } 2} + (X_2)^2 * \theta_{\text{var}} + M \quad (10)$$

Onde X_1 é a média da RSSI, X_2 é o desvio padrão, $(X_2)^2$ é a variância, uma vez que o quadrado do desvio padrão é igual à variância, Y_{ex} é o número estimado de pessoas de cada modelo, o $\theta_{\text{média}}$ é o parâmetro a serem estimado de acordo com a RSSI média, o θ_{var} é o parâmetro a ser estimado de acordo com a variância, o θ_{desvp} é o relacionado com o desvio padrão, o $\theta_{\text{média } 2}$ é o parâmetro a ser estimado de acordo com a RSSI média quadrática e o $\theta_{\text{constante}}$ é o parâmetro relacionado à constante.

Estes modelos também podem ser vistos na forma matricial, nas Equações 11, 12 e 13.

$$1^\circ \text{Ordem: } Y_{e1} = \begin{bmatrix} X_1 \\ X_2^2 \\ 1 \end{bmatrix} * [\theta_{\text{média}} \ \theta_{\text{var}} \ \theta_{\text{constante}}] \quad (11)$$

$$2^\circ \text{Ordem: } Y_{e2} = \begin{bmatrix} X_1 \\ X_1^2 \\ X_2 \\ X_2^2 \\ 1 \end{bmatrix} * [\theta_{\text{média}} \ \theta_{\text{média } 2} \ \theta_{\text{desvp}} \ \theta_{\text{var}} \ \theta_{\text{constante}}] \quad (12)$$

$$3^\circ \text{Ordem: } Y_{e3} = \begin{bmatrix} X_1 \\ X_1^2 \\ X_2^2 \\ 1 \end{bmatrix} * [\theta_{\text{média}} \ \theta_{\text{média } 2} \ \theta_{\text{var}} \ \theta_{\text{constante}}] \quad (13)$$

4. METODOLOGIA

São descritos neste capítulo o ambiente de testes, equipamentos utilizados, método da calibração dos sensores e o firmware utilizado nos sensores.

4.1. Ambiente de Testes e Equipamentos Utilizados

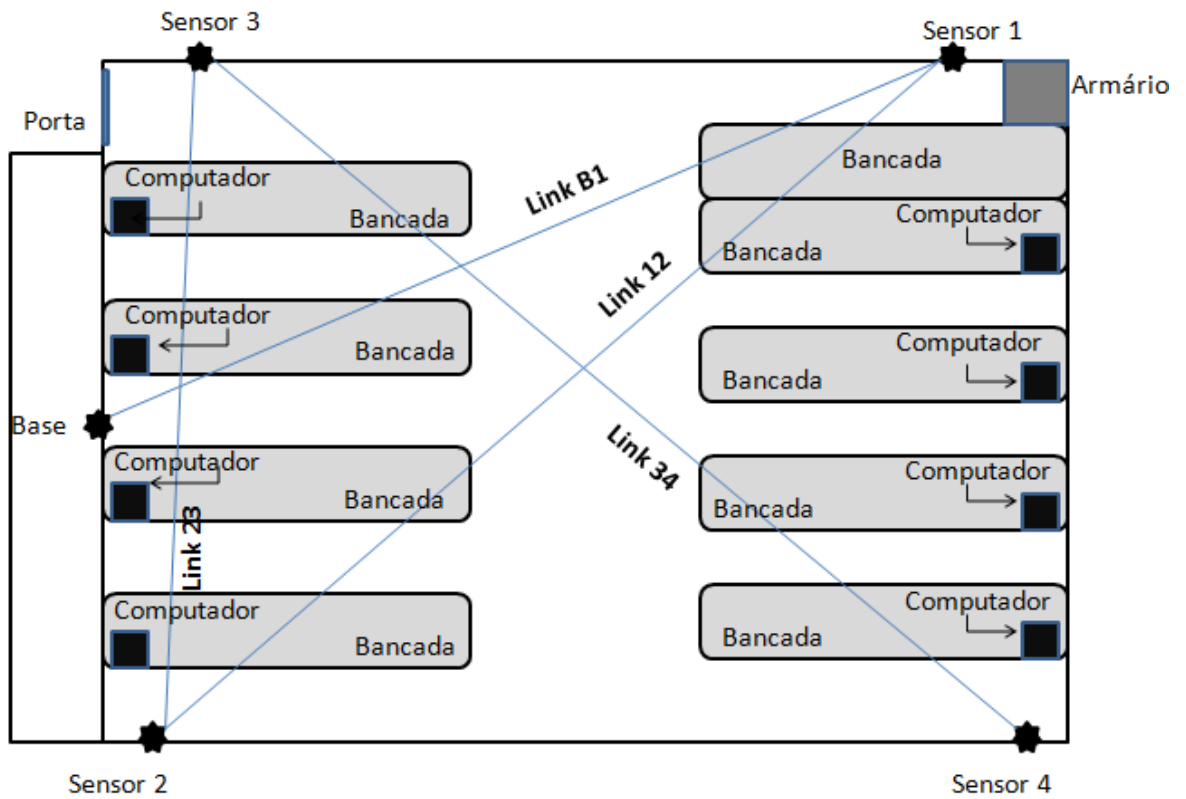
Os testes para a coleta da RSSI para análise posterior foram feitos em uma sala em que se encontra um laboratório de meios de transmissão na Puc-Campinas.

Este laboratório foi escolhido devido a uma alta rotatividade do número de pessoas em um ambiente fechado e pelo fato de se encontrar dentro da Universidade em que o trabalho foi realizado.

Para a coleta dos dados foi montada uma rede com uma base e quatro nós sensores.

Cada nó sensor foi posto em cada um dos 4 cantos da sala a uma altura de 1,3 metros. A base, também na altura de 1,3 metros, se encontrava em posição estratégica para sua ligação com o notebook. Esta altura foi utilizada com base na característica do ambiente de testes e faz com que as pessoas sentadas influenciem nos links, atuando como obstrução nos sinais, e as bancadas não influenciem. Esta altura também foi utilizada em BRAGA (2014) e para efeito de comparação, ela foi mantida. Na Figura 2 é apresentada a vista por cima do laboratório com a disposição da base e dos nós sensores. A altura de 1,3 metros foi escolhida para que a movimentação de pessoas influencie na RSSI. De acordo com pesquisa do IBGE de 2009, a estatura média do Brasileiro é de 1,72 metros.

Figura 2 - Layout vista de cima do ambiente de testes



Na Figura 3 é ilustrado o laboratório que foi utilizado como ambiente de testes.

Figura 3 - Laboratório onde foram realizados os experimentos



A base foi conectada em um computador Acer Aspire com processador Core i5 2.3 GHz, Memória de 4GB DDR3, com sistema operacional Windows 7, através de um cabo USB (Universal Serial Bus).

A base é composta por um módulo UART, ilustrado na Figura 4 e um modulo BE900, ilustrado na Figura 5. Ambas as figuras foram extraídas de (RADIUINO, 2015).

Figura 4 - Modulo UART



Figura 5 - BE900



O Módulo UART é utilizado na programação do Firmware dos rádios, tanto na base como nos nós sensores e também é utilizado na interface entre a base e o computador, por meio de USB.

Tanto na base como nos nós sensores foi utilizada uma antena monopolo vertical de 10 centímetros de altura e com ganho aproximado em 3 dBi. A Figura 6 ilustra a antena utilizada.

Figura 6 - Antena monopolo vertical



Nos nós sensores, foram utilizadas placas do kit DK104 (ilustrado na Figura 7) o módulo BE900 e antenas monopolo vertical iguais à utilizada na base.

Figura 7 - Placa do kit DK104



O módulo BE900 é composto por um processador AVR Atmega328 e por um transceptor TI CC1101 e opera na banda não licenciada ISM (*Industrial, Scientific and Medical*) em 915 MHz (RADIOIT, 2015). Com este módulo, é possível fazer a monitoração e o controle de grandezas (RADIOIT, 2015). Ele atende à regulamentação ANATEL e FCC. Abaixo, seguem as características do BE900:

- É compatível com a plataforma Arduino, podendo utilizar o IDE Arduino para gravação do firmware.
- Conector SMA 90° fêmea.
- Microcontrolador: AVR Atmega 328 de 8 bits, 32k de ROM, 2k de RAM, 1k de EEPROM e clock de 8 MHz.
- Transceptor TI CC1101 que opera em 915 MHz (902-907,5 MHz e 915-928 MHz).

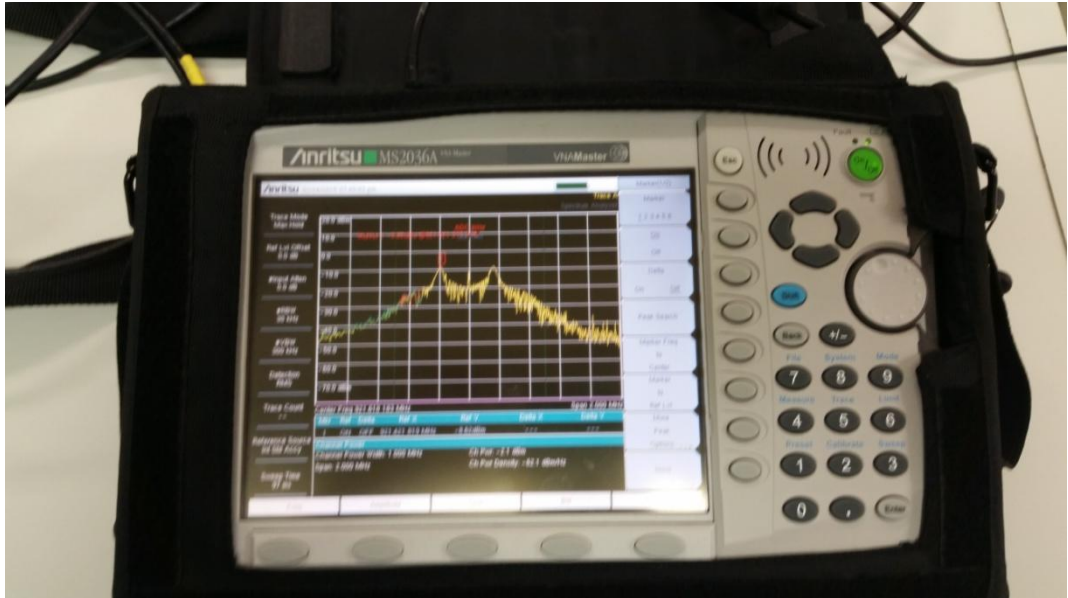
Características do sistema rádio (Módulos BE900):

- Frequência de operação: 915 MHz
- Modulação: 2-FSK – Modulação padrão utilizada no 802.15.4.
- Potência de transmissão: Até 10 mW (+10dBm). Potência máxima foi utilizada nos testes.
- Taxa de dados: De 4.8 kbps até 250 kbps. Nos testes foi utilizada a taxa mínima de 4.8 kbps

4.2. Calibração dos rádios

Para uma maior precisão e confiança dos dados coletados na execução dos testes, é necessário que haja uma aferição do medidor de potência da base e dos nós sensores. O medidor de potência da base e dos nós sensores foram aferidos utilizando um analisador de espectro da Anritsu modelo MS2036A. A Figura 8 ilustra o analisador de espectro utilizado.

Figura 8 - Analisador de Espectro utilizado na calibração dos sensores



Em primeiro lugar é medida a potência de transmissão dos rádios. Esta fase consiste em ajustar a base com a potência máxima de transmissão, que é de 10 dBm. Após o ajuste é necessário ligar o rádio (base) diretamente no analisador de espectro e medir a potência que é recebida pelo analisador. A potência recebida pelo analisador deve ser a mesma potência transmitida pelo transmissor.

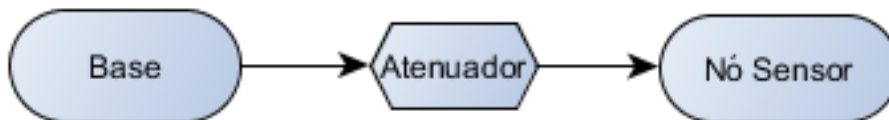
Após o ajuste da potência de transmissão da base, é feita a verificação do medidor de potência de recepção dos rádios. Para esta etapa são utilizados o analisador de espectro, os rádios, um atenuador e cabos. O rádio (devidamente ligado a um computador) deve ser conectado diretamente no atenuador, e o atenuador deve ser conectado no analisador de espectro. A Figura 9 ilustra esta etapa. Deve ser feitos três testes com a potência de transmissão do rádio ajustada em 10 dBm, 0 dBm e -10 dBm respectivamente em cada teste. O atenuador deve ter um ajuste fixo de -30 dBm. É esperado que na saída (no analisador de espectro) a potência de recepção seja de aproximadamente -20 dBm, -30 dBm e -40 dBm respectivamente para cada teste. Todos os valores coletados devem ser anotados para comparação posterior.

Figura 9 - Etapa de aferição do medidor de potência do nó sensor



Na última etapa da calibração, a ligação é feita diretamente entre o rádio transmissor conectado no atenuador, e este último ligado ao rádio receptor. O diagrama da montagem é apresentado na Figura 10.

Figura 10 - Última etapa aferição medidor de potência do nó sensor



Os mesmos valores de transmissão foram utilizados nesta etapa e o mesmo valor do atenuador. Para a coleta da potência de recepção foi utilizado um software desenvolvido em *Python* (Disponível no Anexo A).

4.3. Processo de coleta de dados

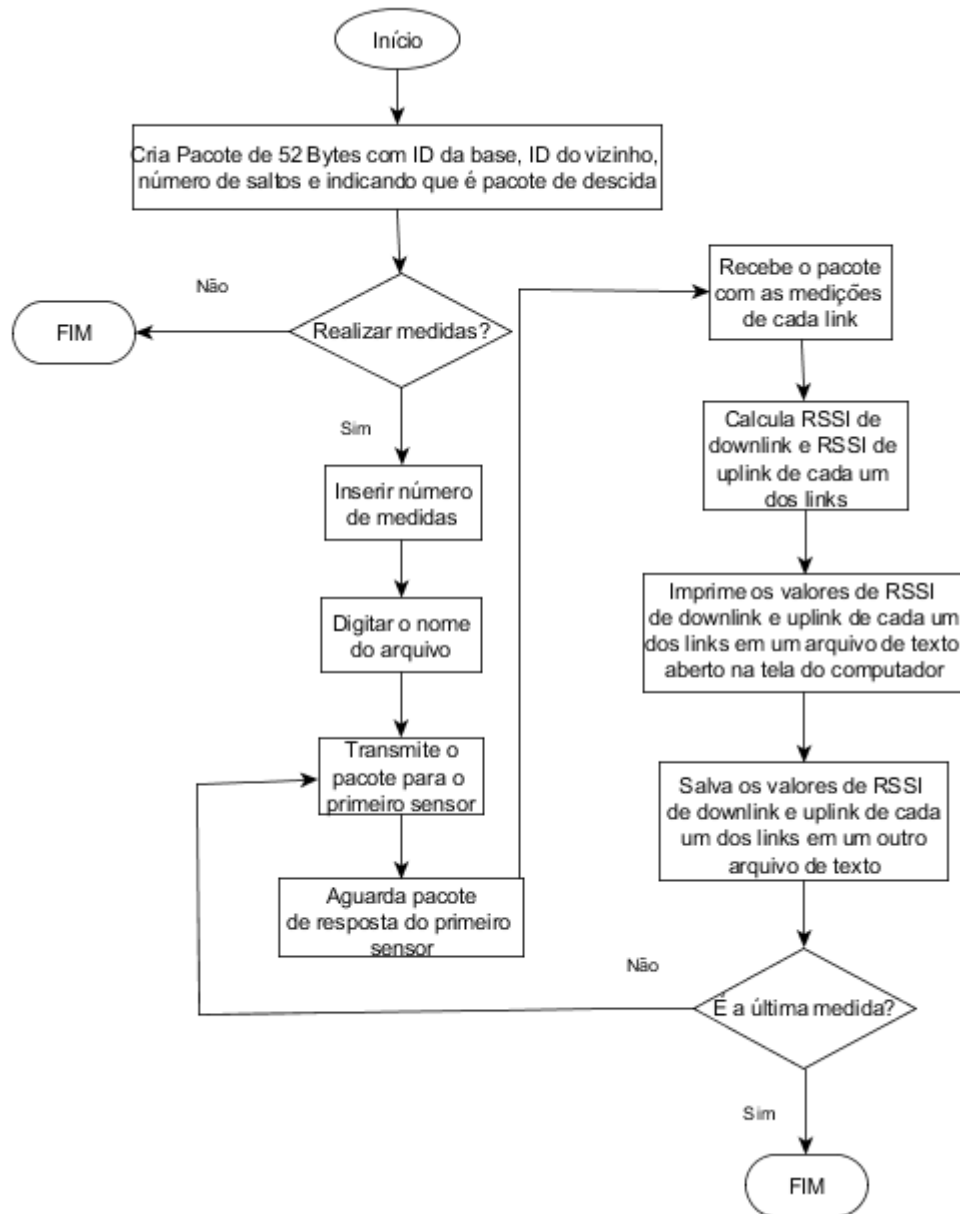
No processo de coleta de dados, primeiramente tanto a base como os nós sensores foram gravados com um firmware que possui um protocolo de roteamento fixo seqüencial de 4 sensores, ou seja a rede possui 4 saltos (será explicado no item 4.3.1).

Após a gravação do firmware e a montagem da rede no ambiente de testes e todos os rádios devidamente ligados, é iniciada a aplicação para coleta de dados com um script desenvolvido em *Python*.

Este script constrói um pacote de 52 Bytes a ser enviado pela base para o nó sensor 1 com a requisição da RSSI na descida do pacote. O mapa do pacote radiuino de 52 Bytes está disponível no Anexo D.

No outro sentido, ou seja, quando o pacote está subindo, a base recebe o pacote do nó sensor 1, o script *Python* recebe esse pacote, calcula as RSSI's de todos os links gravadas no pacote e gera um arquivo de texto com todos estes dados. Na Figura 11 é possível observar o funcionamento do script de coleta de dados.

Figura 11 - Fluxograma de funcionamento do script de coleta de dados python



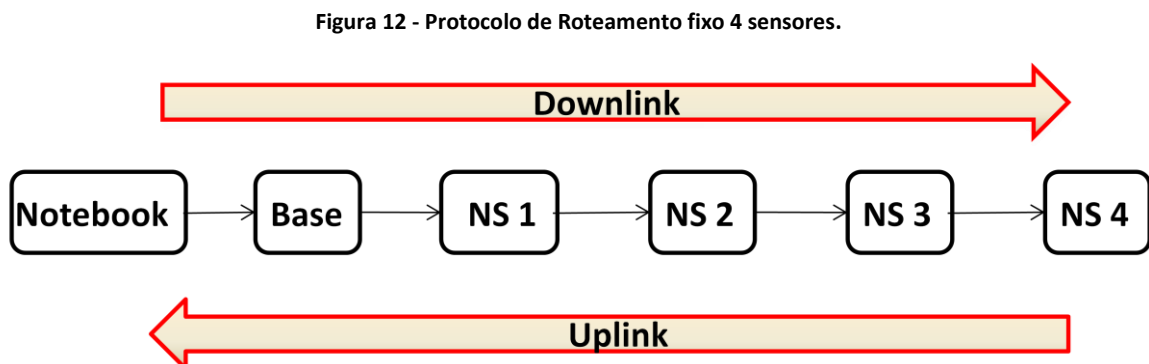
O acompanhamento dos testes foi feito em tempo real para anotação do número real de pessoas que estiveram no ambiente de testes e o intervalo de tempo de cada número de pessoas.

- O Item 4.3.1, explica o funcionamento deste protocolo.
- O Item 4.3.2, explica como foi feito o tratamento dos dados.

4.3.1. Protocolo de roteamento com quatro sensores

Este protocolo de roteamento foi utilizado por sua fácil aplicação e entendimento e também pelo fato de que neste trabalho ficou decidido que seria interessante tratar a RSSI de links em que ficassem nas diagonais da sala em que os testes foram feitos. Com este protocolo foi possível a coleta deste tipo de dado.

O funcionamento do protocolo é ilustrado na Figura 12:



A rede é composta por uma base, diretamente ligada a um notebook e quatro nós sensores.

Para um melhor entendimento é necessário a introdução de alguns conceitos utilizados neste protocolo de roteamento.

Primeiramente este protocolo define algumas situações para o roteamento dos pacotes, são elas:

- O pacote está descendo e não chegou ao último sensor da corrente (sentido *downlink*).
- O pacote está descendo e chegou ao último sensor da corrente.
- O pacote irá subir, pois já alcançou o último sensor da rede.

Cada sensor possui sua identificação (ID), a identificação do nó vizinho na descida e a identificação do nó destino na sua subida.

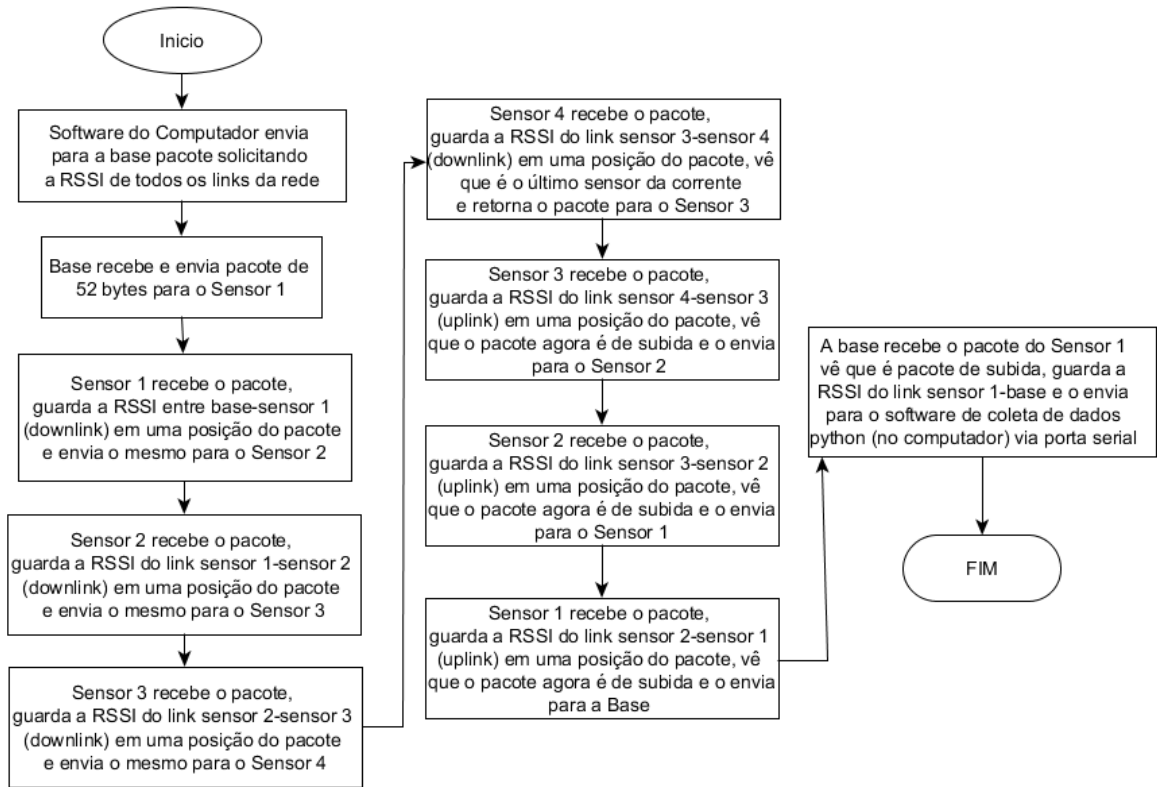
Quando o pacote está descendo, ou seja, está no sentido de *downlink*, o próximo sensor da corrente de sensores é o nó vizinho. Por exemplo, na Figura 8 o nó vizinho do sensor NS 1 é o NS 2.

Quando o pacote está subindo, ou seja, está no sentido de *uplink*, o próximo sensor é o destino. Por exemplo, na Figura 12 o nó destino do sensor NS 3 é o NS 2.

O computador, através de um script desenvolvido em *Python* (disponível no Anexo A) e instalado no computador, envia uma mensagem requisitando a RSSI dos sensores. A base envia esta mensagem para o nó sensor 1 (NS 1) com esta requisição, o NS 1 verifica a RSSI do link Base-NS 1 e guarda em uma posição do pacote. O mesmo acontece em cada link, NS1-NS2, NS2-NS3 e NS3-NS4. Estas RSSI's coletadas são de *downlink*. Após o pacote chegar ao último sensor da corrente, o mesmo volta requisitando a RSSI de *uplink* de cada link. Desta vez os links são: NS4-NS3, NS3-NS2, NS2-NS1 e NS1-Base.

O fluxograma da Figura 13 ilustra o procedimento do protocolo de roteamento utilizado.

Figura 13 - Fluxograma do protocolo de roteamento.



RSSI de *Downlink* pode ser definida como a potência de recepção de um nó sensor quando o mesmo recebe o pacote do seu sensor destino, no caso da Figura 12, seria a potência de recepção medida no NS 2 quando ele recebe um pacote do NS 1. *Downlink* seria a descida do pacote.

Já a RSSI de *Uplink* é a potência de recepção de um nó sensor quando o mesmo recebe um pacote de seu sensor vizinho. No caso da Figura 12 seria a potência de recepção medida no sensor NS 3 quando ele recebe um pacote do sensor NS 4.

No anexo B se encontra o código do protocolo de roteamento utilizado nos sensores.

4.3.2. Tratamento dos dados

Para o tratamento dos dados, o software de coleta de dados em *Python* gera um arquivo log em “txt” com as RSSI’s de cada link separado assim como o horário em que aquele dado (RSSI no caso) foi coletado.

Utilizando o software Microsoft Excel, os dados do log foram inseridos em planilhas para sua posterior análise.

Primeiramente foi feita a separação de cada janela com o número de pessoas que continha na sala, numa janela de tempo específica em que se obteve cada número de pessoas na sala e que as mesmas permanecessem estáveis.

Em cada janela (ou grupo de pessoas), a RSSI que inicialmente foi medida em dBm, teve de ser transformada para milliWatt. Essa mudança se deve à premissa de que não se deve tirar média diretamente em dBm, uma vez que este tipo de escala é da forma logarítmica e não linear.

É importante salientar que em Braga (2014), diferentemente deste trabalho, esta média foi feita diretamente em dB, aproveitando a distribuição log-normal e o fading para o cálculo da média diretamente em dB.

Para a conversão de dados da escala em dBm para milliWatt é necessário a utilização da Equação 14.

$$P_{mw} = 10^{(P_{dBm} / 10)} \quad (14)$$

Após a conversão citada acima, a média da RSSI de cada link foi calculada.

O tratamento dos dados são feitos em dBm, a conversão para Watt foi feita somente para que fosse possível tirar a média da RSSI de cada link.

Para converter uma potência em milliWatt para dBm, basta aplicar a Equação 15.

$$P_{dBm} = 10 * \log_{10} P_{milliwatt} \quad (15)$$

5. RESULTADOS

Todos os resultados obtidos de cada link e da média geral dos links serão apresentados neste capítulo, assim como o melhor modelo obtido para estimação da taxa de ocupação do ambiente interno utilizado nos testes.

5.1. Processamento dos dados

Após o tratamento inicial dos dados comentado no item 4.3.2, os dados coletados nos testes foram separados em índices pares e ímpares. Esta é uma estratégia utilizada para fazer a modelagem dos dados, que consiste em separar em um bloco de dados os de índice par (linhas de índice par) e em outro bloco os dados de índice ímpar (as linhas de índice ímpar). Os dados de índices pares foram utilizados como dados de modelagem para a estimação dos parâmetros (estes parâmetros são os valores encontrados no cálculo do vetor Θ).

Os dados de índice ímpar foram utilizados como dados de validação. Os dados de validação são utilizados para validar os modelos e encontrar a saída estimada de cada modelo. No caso deste trabalho a saída estimada desejada é o número estimado de pessoas no ambiente em que foram efetuados os testes.

Além da média da RSSI de cada link, o desvio padrão, a variância e o quadrado da RSSI média, calculada diretamente em dB, também foram calculados para serem utilizados como valores de entrada para a estimação dos parâmetros (índices pares) e validação dos modelos (índices ímpares).

Após a separação dos índices dos dados, foram gerados arquivos de texto para cada link com a finalidade de processamento final dos dados no software de tratamento de dados Scilab. Todos os valores foram normalizados e são mostrados em tabelas dentro da sessão de cada link, a seguir. A normalização foi feita para o condicionamento dos dados. Dados normalizados são considerados em um intervalo entre 0 e 1. A Equação 16 foi utilizada para a normalização.

$$\text{Num}_{\text{norm}} = (N_1 - N_{\text{min}}) / (N_{\text{max}} - N_{\text{min}}) \quad (16)$$

Onde:

Num_{norm} : Número normalizado;

N_1 : Número (valor) a ser normalizado;

N_{min} : Valor mínimo entre os valores a serem normalizados;

N_{max} : Valor máximo entre os valores a serem normalizados.

Na etapa seguinte foi feita a identificação de outliers de cada link utilizando somente os dados de modelagem. Outliers são dados que apresentam um comportamento diferente dos outros dados, ou seja, um ou mais dados que apresentam um comportamento fora do padrão.

Este processo foi efetuado para cada link de forma unitária e também para a média geral dos links.

5.2. Escolha do modelo

Neste item serão apresentados todos os passos para a escolha do modelo final, de acordo com os passos citados no item 3.2. A escolha do modelo será feita para cada link de forma individual e também para a média dos links, ou seja, foi calculada a média geral da RSSI de todos os links para serem tratados de forma única.

5.2.1. Link Base – Sensor 1

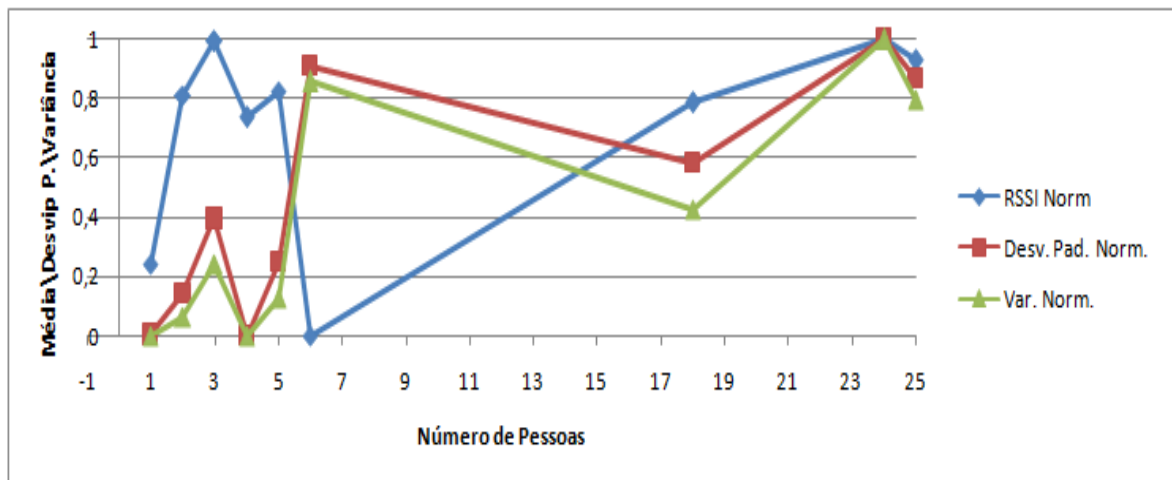
Após o tratamento inicial dos dados, foi construída a Tabela 1 normalizada, com a média, desvio padrão, variância e média quadrática para cada número de pessoas. Estes dados são os dados de modelagem.

Tabela 1 - Média, Desvio Padrão, Variância e Média Quadrática normalizados do Link Base – Sensor 1.

Número de Pessoas	RSSI média	Desvio Padrão	Variância	RSSI média quadrática
1	0,24106208	0,00327835	0,00117259	0,74652958
2	0,81367625	0,14297446	0,06401016	0,17604126
3	0,9932781	0,39534433	0,24129416	0,00626906
4	0,73664671	0	0	0,25019569
5	0,82733006	0,25120663	0,12998738	0,16298104
6	0	0,91061473	0,85816066	1
18	0,7860935	0,58399952	0,42743806	0,20250199
24	1	1	1	0
25	0,92850696	0,86997822	0,79708234	0,06699081

Com os dados da Tabela 1 foi construído o gráfico da Figura 14 para a identificação de possíveis *outliers*.

Figura 14 - Gráfico RSSI Média, Desvio Padrão e Variância normalizados do link Base – Sensor 1



É interessante ressaltar que neste link a RSSI variou muito, não apresentando um padrão, principalmente entre 1 e 6 pessoas de forma a não ser possível a identificação e remoção de *outliers*.

A variação deste link pode ser justificada por uma possível obstrução de alguma pessoa em um dos sensores do link ou em ambos os sensores.

Após a identificação de *outliers* a próxima etapa é a estimação dos parâmetros de cada modelo, de 1°, 2° e 3° ordens. Para tal estimação foi utilizado um script desenvolvido no software Scilab (Disponível no Anexo C).

Como dados de entrada foram utilizados os dados de modelagem (dados de índice par).

Os parâmetros estimados se encontram na Tabela 2.

Tabela 2 - Parâmetros estimados para os modelos do link: Base - Sensor 1

Parâmetro	1° ordem	2° ordem	3° ordem
Θ (média)	11.012117	- 1111.6451	- 823.75858
Θ (variância)	18.210029	- 0.0990853	26.634361
Θ (constante)	- 5.6273986	1103.2631	819.9916
Θ (média2)		- 1121.7045	- 834.65422
Θ (desvioP.)		38.099897	

Com os parâmetros estimados, os mesmos foram substituídos nas equações de cada modelo, propostos no Capítulo 3, e com a ajuda de um script elaborado no software Scilab foram estimados o número de pessoas. Os dados de entrada utilizados nesta etapa foram os dados de validação (dados de índice ímpar).

Após rodar o script no Scilab, o número de pessoas para o link B1 foi estimado e segue na Tabela 3.

Tabela 3 - Número estimado de pessoas em cada modelo para o link: Base - Sensor 1

Número real de pessoas	1° ordem	2° ordem	3° ordem
1	- 3.1574316	- 1.8079067	- 1.5207697
2	4.1285262	10.247896	7.0175428
3	9.8847465	12.777043	7.6515789
4	2.0875971	7.2017183	6.3467227
5	5.8223175	14.455024	9.0354911
6	12.58263	17.287432	11.971741
18	14.10846	32.43662	22.681544
24	14.918001	29.666024	10.176594
25	20.367436	33.7078	26.207409

Com os valores da Tabela 3 é possível identificar que houve uma variação grande na estimativa de pessoas e que poucos pontos foram estimados com um erro (resíduo) muito baixo, como em 5 pessoas com o modelo de 1° ordem que estimou em aproximadamente 5,8 pessoas e em 25 pessoas com o modelo de terceira ordem que estimou em aproximadamente 26 pessoas.

Na tabela 4 se encontra o resíduo absoluto médio de cada modelo. Com base neste resíduo será feita a escolha do modelo que melhor estima o número de pessoas.

Tabela 4 - Resíduo absoluto médio de cada modelo do Link: Base - Sensor 1

	1° ordem	2° ordem	3° ordem
RAM	4.4549065	8.1763848	4.9173561

De acordo com o RAM, o modelo de primeira ordem se mostra como o melhor modelo para a estimação do número de pessoas. Mesmo sendo o melhor modelo ele se mostra muito alto levando em consideração o resultado encontrado em BRAGA (2014) que foi de aproximadamente 2 pessoas de erro.

Nos próximos tópicos serão mostrados os resultados dos outros 3 links em que foram efetuados testes e também da média geral dos links para que seja possível chegar a um modelo para a estimação de pessoas.

Na Tabela 5 encontra-se o resíduo absoluto para cada grupo de pessoas utilizando o modelo de menor RAM, o de 1° ordem.

Este resíduo também é importante para verificar o erro entre o valor real do número de pessoas e o valor estimado para cada grupo de pessoas.

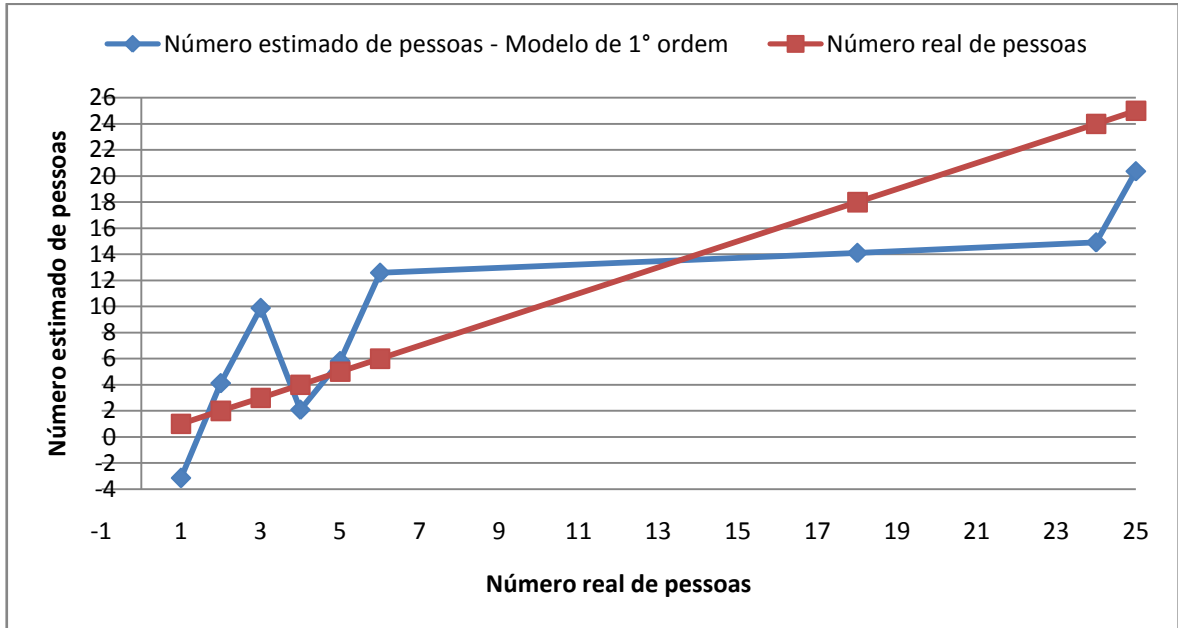
Tabela 5 - Resíduo absoluto com o modelo de 1° ordem do link: Base- Sensor 1

Número real de pessoas	Resíduo absoluto
1	4.1574316
2	2.1285262
3	6.8847465
4	1.9124029
5	0.8223175
6	6.5826304
18	3.8915402
24	9.0819994
25	4.632564

Como mencionado anteriormente os pontos que obtiveram uma melhor estimação foram os de 5 pessoas com um RA menor que uma pessoa.

Na Figura 15 é representada graficamente a diferença entre o número real de pessoas e o número estimado de pessoas pelo modelo de primeira ordem para o link B1.

Figura 15 - Comparação entre número real de pessoas e número estimado de pessoas pelo modelo de 1° ordem



Na Figura 15 observa-se que somente os grupos de 2 e 5 pessoas obtiveram uma estimaco de pessoas mais prxima do nmero real de pessoas.

O modelo final proposto para o link B1  o modelo de 1° ordem que obteve o menor resduo absoluto mdio e  o que segue:

$$Y_{e1} = X_1 * 11.012117 + (X_2)^2 * 18.210029 - 5.6273986 \quad (17)$$

5.2.2. Link: Sensor 1 – Sensor 2

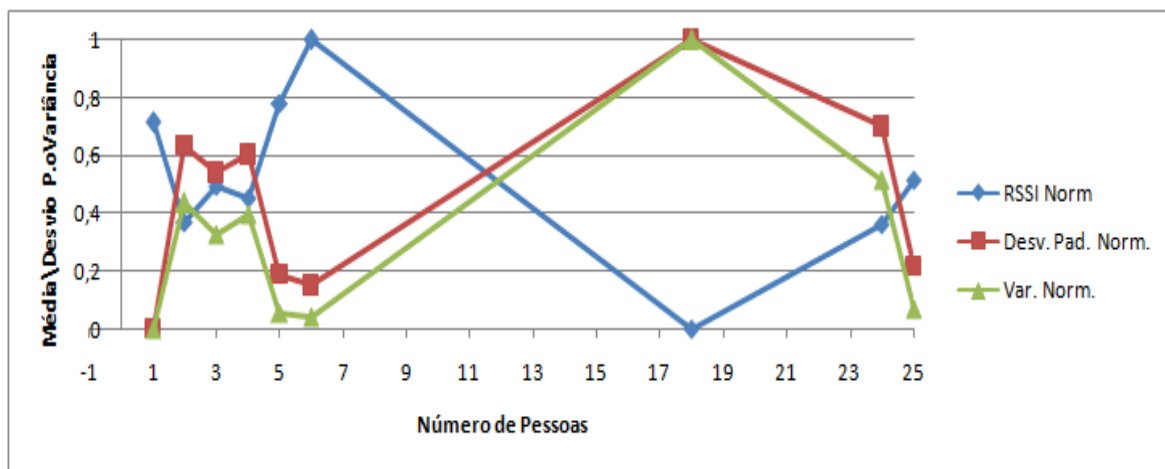
Para o Link 12, foi gerada a Tabela 6 com as mesmas entradas do link anterior.

Tabela 6 - Média, Desvio Padrão, Variância e Média Quadrática normalizados do link Sensor 1 - Sensor 2

Número de Pessoas	RSSI média	Desvio Padrão	Variância	RSSI média quadrática
1	0,71500225	0	0	0,26803997
2	0,36855531	0,63528807	0,43480905	0,61207788
3	0,49319837	0,54117547	0,32632664	0,48600085
4	0,45454674	0,60314293	0,39603219	0,52482056
5	0,77998888	0,1880904	0,05595397	0,2057303
6	1	0,15122264	0,04016233	0
18	0	1	1	1
24	0,36115947	0,69966135	0,51783895	0,61964007
25	0,51655232	0,2158693	0,0694064	0,46266585

Com base nos dados da Tabela 6 foi gerado o gráfico da Figura 16 que mostra a variação da Média, Desvio Padrão e Variância.

Figura 16 - Gráfico RSSI Média, Desvio Padrão e Variância normalizados do link Sensor 1 – Sensor 2



Neste link também foi observada uma variação grande dos dados de entrada, principalmente entre 1 e 6 pessoas.

Entre 6 e 18 pessoas é possível notar uma diminuição da média e aumento do desvio padrão e variância.

Com base nesta observação poderiam ser identificados como *outliers* os pontos de 1, 24 e 25 pessoas. Porém como este link apresentou uma grande variação entre os pontos, nenhum *outlier* foi identificado e retirado.

O cálculo dos parâmetros foi feito sem a retirada de *outliers*, como mencionado anteriormente, utilizando os dados de modelagem.

A Tabela 7 contém os valores dos parâmetros estimados após utilização do software Scilab.

Tabela 7 - Parâmetros estimados para o link: Sensor 1 - Sensor 2

Parâmetro	1° ordem	2° ordem	3° ordem
Θ (média)	- 22.895118	1124.3584	505.94757
Θ (variância)	- 8.1522024	- 114.35279	- 36.23422
Θ (constante)	23.92224	- 1122.6439	- 502.22351
Θ (média2)		1203.2366	558.53777
Θ (desvioP.)		52.602347	

Após a estimação dos parâmetros, os mesmos foram utilizados para a estimação do número de pessoas tendo como dados de entrada os dados de validação (índice ímpar).

Na Tabela 8 está o número estimado de pessoas para cada modelo após a rotina no software Scilab ser utilizado para tal.

Tabela 8 - Número estimado de pessoas de cada modelo para o link: Sensor 1 - Sensor 2

Número real de pessoas	1° ordem	2° ordem	3° ordem
1	7.6168426	3.7788254	9.2803402
2	10.792007	1.3143095	5.2595872
3	9.446486	2.6818473	4.3514457
4	9.6653071	1.0002551	3.8736481
5	5.5527161	4.9175764	4.7469324
6	0.6477223	4.8378424	2.0377365
18	15.770038	18.842257	20.08004
24	14.196399	28.470496	20.201797
25	11.538919	17.970733	14.757776

O resíduo absoluto médio foi calculado para cada modelo e se encontra na Tabela 9.

Tabela 9 - Resíduo absoluto médio dos três modelos para o link: Sensor 1 - Sensor 2

	1 ordem	2 ordem	3 ordem
RAM	6.5466978	2.2632239	3.705947

De acordo com o RAM da Tabela 9 o modelo de segunda ordem se mostrou o melhor para o link 12. Apesar deste resíduo não muito grande (ele se aproxima do RAM encontrado em BRAGA (2014) é possível verificar na Tabela 10 que o resíduo absoluto do modelo de 2º ordem de cada grupo de pessoas foi muito alto em alguns casos (devido à variação da média e do desvio padrão vistos no gráfico da Figura 14).

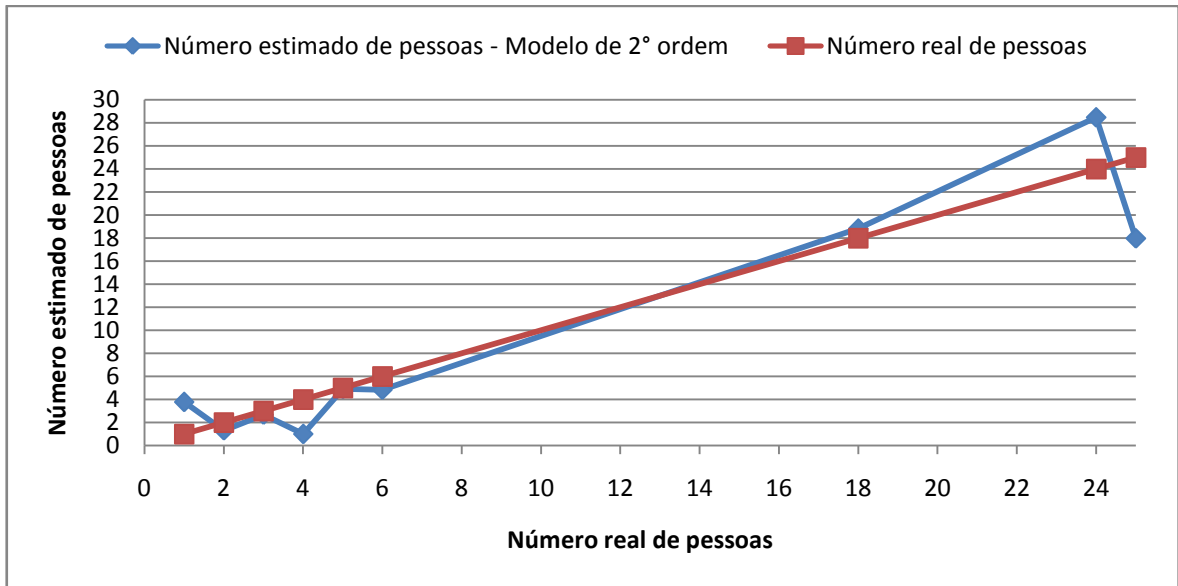
Tabela 10 - Resíduo absoluto encontrado utilizando o modelo de 2º ordem para o link: Sensor 1 - Sensor 2

Número real de pessoas	Resíduo absoluto
1	2.7788254
2	0.6856905
3	0.3181527
4	2.9997449
5	0.0824236
6	1.1621576
18	0.842257
24	4.4704958
25	7.0292673

Os pontos em destaque com maior RA são os de 24 e 25 pessoas. Porém em termos percentuais o grupo de 4 pessoas apresenta o maior resíduo absoluto com um percentual de erro de aproximadamente 75%.

O gráfico da Figura 17 apresenta a diferença visual entra a reta ideal, que é o número real de pessoas, e o número estimado de pessoas.

Figura 17 - Comparação entre número real de pessoas e número estimado pelo modelo de 2° ordem - Link 12



No gráfico da Figura 17 os pontos que se destacam são os de 24 e 25 pessoas que possuem um erro maior com relação aos outros pontos.

Com base no menor resíduo absoluto médio, o modelo escolhido é o de 2° ordem e é mostrado a seguir.

$$Y_{est2} = X_1 * 1124.3584 + X_1^2 * 1203.2366 + X_2 * (52.602347) + X_2^2 * (-114.35279) - 1122.6439 \quad (18)$$

Onde:

Y_{est2} : é o número estimado de pessoas do modelo de 2° ordem.

X_1 : RSSI média.

X_2 : é o desvio padrão.

X_2^2 : é a variância.

5.2.3. Link: Sensor 2 – Sensor 3

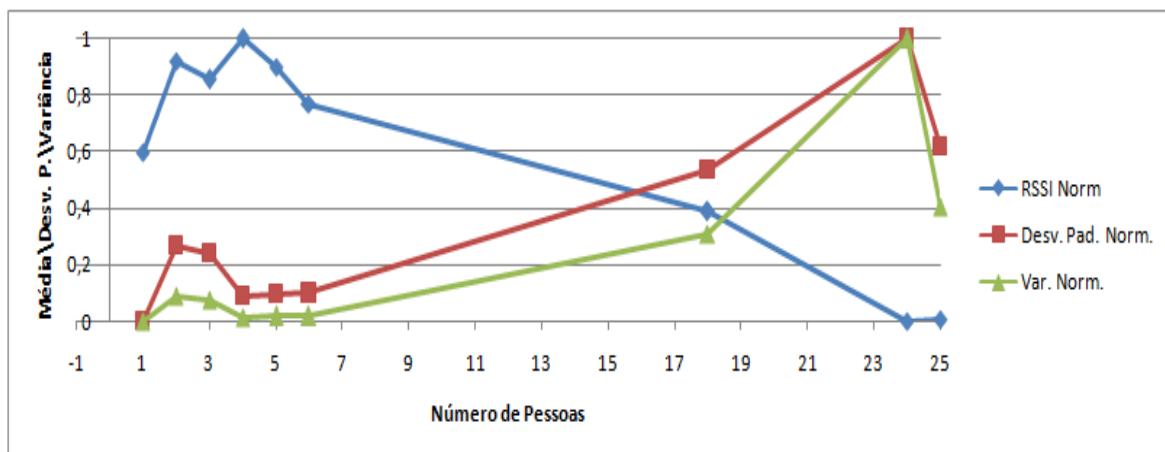
Na Tabela 11 estão os dados de modelagem normalizados do link 23.

Tabela 11 - Média, Desvio Padrão, Variância e Média Quadrática normalizados do link Sensor 2 - Sensor 3

Número de Pessoas	RSSI média	Desvio Padrão	Variância	RSSI média quadrática
1	0,5980492	0	0	0,37507285
2	0,92118854	0,26921472	0,09047738	0,07069392
3	0,85513947	0,24233083	0,07552356	0,13100978
4	1	0,08975285	0,01553057	0
5	0,901562	0,0955149	0,01702763	0,08851494
6	0,76962682	0,10128867	0,01858824	0,21054886
18	0,38969912	0,53485075	0,30882828	0,58370835
24	0	1	1	1
25	0,00723184	0,6186264	0,40428514	0,99196541

O gráfico da Figura 18 ilustra a média, desvio padrão e variância, todos normalizados, do link 23 que foi elaborado a partir dos dados da Tabela 11.

Figura 18 - Gráfico RSSI Média, Desvio Padrão e Variância normalizados do link Sensor 2 – Sensor 3.



É possível observar uma tendência no link 23 de diminuição da média da RSSI à medida que o número de pessoas aumenta.

Já o desvio padrão e a variância tendem a aumentar à medida que o número de pessoas aumenta.

Foram observados dois *outliers* neste link. O ponto 1 mostra uma RSSI muito distante dos outros pontos próximos a ele. Já o ponto 25 se mostra fora da tendência de aumento do desvio padrão/variância.

Ambos os pontos foram retirados para cálculo posterior dos parâmetros e da identificação do modelo para a estimação de pessoas.

O próximo passo é o cálculo dos parâmetros. A mesma forma utilizada nos links anteriores é feita aqui. Os dados de modelagem (índice par) são utilizados para estimação dos parâmetros e para isso é utilizado um script de estimação de parâmetros construído no Scilab.

Na Tabela 12 estão os parâmetros calculados pelo script para cada modelo para o link 23.

Tabela 12 - Parâmetros estimados para cada um dos modelos propostos – Link Sensor 2–Sensor 3.

Parâmetro	1° ordem	2° ordem	3° ordem
$\Theta(\text{média})$	- 26.959756	773.77342	376.07679
$\Theta(\text{variância})$	- 4.2039438	- 88.457205	- 38.321896
$\Theta(\text{constante})$	28.403611	- 770.3962	- 372.20288
$\Theta(\text{média}^2)$		856.65116	434.99367
$\Theta(\text{desvioP.})$		26.223891	

Com o cálculo dos parâmetros foi utilizada outra rotina no script Scilab que faz a estimação do número de pessoas.

Os parâmetros e os dados de validação (índices ímpares) foram utilizados para tal estimação. O número de pessoas estimado para cada um dos três modelos pode ser visto na Tabela 13.

Tabela 13 - Número estimado de pessoas para cada modelo - Link: Sensor 2 - Sensor 3.

Número real de pessoas	1° ordem	2° ordem	3° ordem
2	3.0848899	0.7366390	0.4955117
3	4.9746275	3.0384658	3.3512564
4	1.3807845	4.3090491	3.2989782
5	3.9997003	4.0177921	4.6717289
6	7.6044592	6.560935	8.0797772
18	16.990019	19.519802	17.52303
24	23.248709	17.84799	20.74026

Para este link o resíduo absoluto médio para cada modelo pode ser visto na Tabela 14.

Tabela 14 - Resíduo absoluto médio dos três modelos - Link: Sensor 2 - Sensor 3.

	1° ordem	2° ordem	3° ordem
RAM	1.4349663	1.5465473	1.2430751

Neste link o modelo que possui o menor RAM é o de terceira ordem com um erro um pouco maior de uma pessoa, portanto este modelo é o que mais se aproxima do número real de pessoas utilizando somente o link 23 para a estimação de pessoas.

Percebe-se também que os três modelos obtiveram um RAM muito próximo.

Na Tabela 15 encontra-se o resíduo absoluto calculado para cada grupo de pessoas.

Tabela 15 - Resíduo absoluto encontrado para cada grupo de pessoas.

Número real de pessoas	Resíduo Absoluto
2	1.5044883
3	0.3512564
4	0.7010218
5	0.3282711
6	2.0797772
18	0.4769705
24	3.2597405

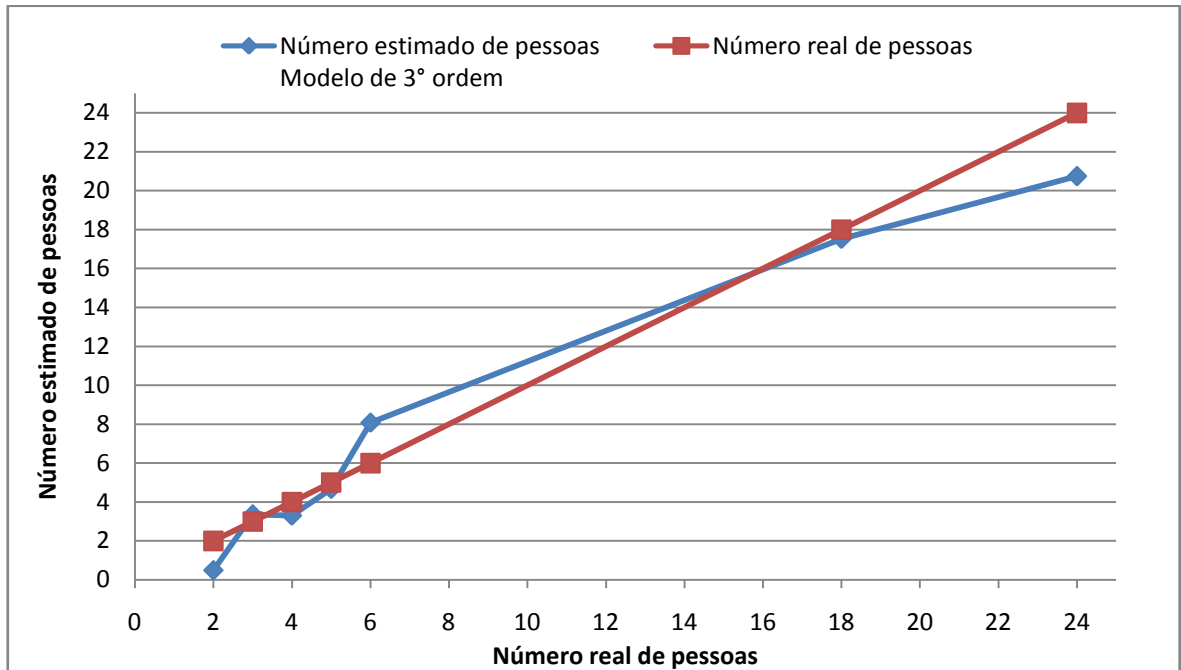
Na Tabela 15 é possível verificar que o maior resíduo absoluto calculado foi para o grupo de 24 pessoas com um RA de aproximadamente 3.26 pessoas aproximadamente.

Entretanto este erro representa menos que 14% do número de pessoas que existiam no ambiente.

No grupo de 2 pessoas, com um RA de 1.5 pessoas, chega-se ao erro percentual de aproximadamente 75%. Este é o único ponto mais discrepante na estimação de pessoas.

O gráfico da Figura 19 faz uma comparação entre o número real de pessoas e o número estimado de pessoas utilizando o modelo de 3° ordem para o link 23.

Figura 19 - Comparação entre número real de pessoas e número estimado de pessoas para o link 23.



Graficamente também é possível observar a aproximação entre o número estimado de pessoas pela modelagem matemática e o número real de pessoas.

O modelo final, de terceira ordem, para o link 23 fica como segue:

$$Y_{e3} = X_1 * 376.07679 + (X_1)^2 * 434.99367 + (X_2)^2 * (- 38.321896) - 372.20288 \quad (19)$$

Onde:

Y_{e3} : é o número estimado de pessoas do modelo de 3° ordem.

X_1 : RSSI média.

X_2^2 : é a variância.

5.2.4. Link: Sensor 3 – Sensor 4

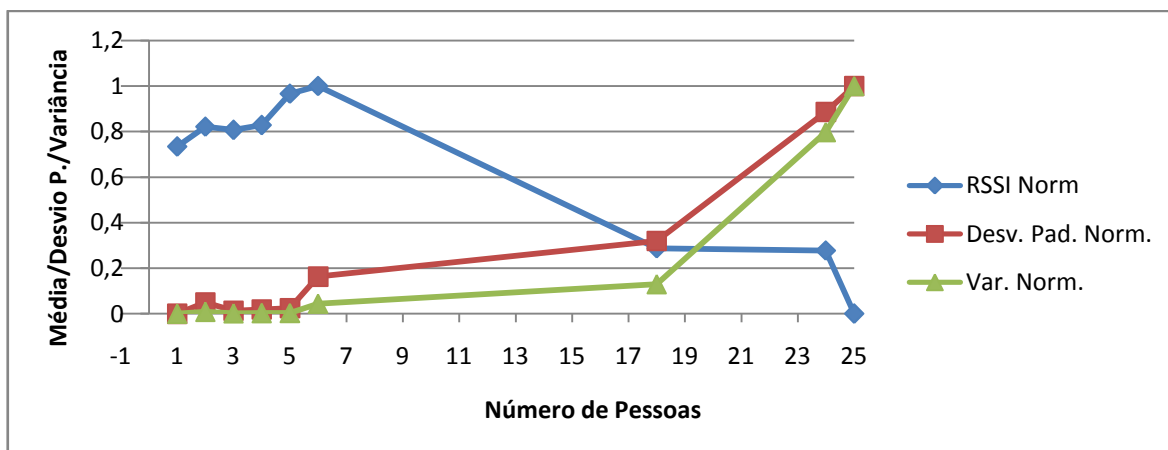
Segue a Tabela 16 com os valores dos dados de modelagem normalizados do link 34.

Tabela 16 - Média, Desvio Padrão, Variância e Média Quadrática normalizados do link Sensor 3 - Sensor 4.

Número de Pessoas	RSSI média	Desvio Padrão	Variância	RSSI média quadrática
1	0,7337015	0	0	0,24691261
2	0,82111378	0,04916436	0,00837443	0,16431225
3	0,80689879	0,01351552	0,00188176	0,17764149
4	0,8279455	0,01922536	0,00277252	0,15792049
5	0,96615126	0,02468162	0,00367689	0,03060398
6	1	0,16318385	0,04403101	0
18	0,28704755	0,31968651	0,1299152	0,69264702
24	0,27753154	0,88592837	0,79774769	0,70257415
25	0	1	1	1

Com base nos dados da Tabela 16 foi construído o gráfico da Figura 20, que permite observar o comportamento dos dados e identificação de *outliers*.

Figura 20 - Gráfico RSSI Média, Desvio Padrão e Variância normalizados do link Sensor 3 – Sensor 4.



Para este link nenhum *outlier* foi identificado, pois como observado no gráfico da Figura 20, não é possível observar nenhum ponto discrepante entre os dados.

Neste link também é possível verificar a tendência de queda da RSSI média de acordo com o aumento do número de pessoas e, inversamente, o aumento do desvio padrão/variância com o aumento do número de pessoas.

A próxima etapa foi estimar os parâmetros de cada modelo utilizando os dados de modelagem (índice par) no script criado no Scilab.

Após executar o Scilab com os dados mencionados, os seguintes valores foram obtidos para cada modelo, indicados na Tabela 17.

Tabela 17 – Parâmetros estimados para cada um dos modelos propostos – Link Sensor 3–Sensor 4.

Parâmetro	1° ordem	2° ordem	3° ordem
$\Theta(\text{média})$	- 14.36944	36.876001	51.625168
$\Theta(\text{variância})$	11.309546	- 31.743161	9.6623914
$\Theta(\text{constante})$	16.412323	- 35.757904	- 49.055153
$\Theta(\text{média}^2)$		45.32496	67.757289
$\Theta(\text{desvioP.})$		47.748361	

Com os parâmetros calculados da Tabela 17, os mesmos foram substituídos nas equações de cada modelo para o cálculo do valor estimado do número de pessoas utilizando o script de estimação de pessoas criado no Scilab.

Os dados de entrada utilizados agora, média, desvio padrão e variância foram os dados de validação (índice ímpar).

A Tabela 18 indica o número estimado de pessoas calculado para o link 34.

Tabela 18 - Número estimado de pessoas para cada modelo - Link: Sensor 3 - Sensor 4.

Número real de pessoas	1 ordem	2 ordem	3 ordem
1	5.8315153	2.471279	5.5161411
2	4.6816037	4.1559016	4.5264222
3	4.7940162	2.7182769	4.6168736
4	4.5000541	2.6224067	4.375231
5	2.556124	2.4038466	2.9225472
6	2.701835	8.4549511	3.1329954
18	13.766474	17.400401	13.929959
24	22.50754	24.696999	22.383948
25	27.721869	25.572256	28.364527

O resíduo absoluto médio de cada modelo para este link foi calculado e pode ser observado na Tabela 19.

Tabela 19 - Resíduo absoluto médio para cada modelo - Link: Sensor 3 - Sensor 4.

	1° ordem	2° ordem	3° ordem
RAM	2.6663428	1.3562728	2.5588606

Para este link o modelo de segunda ordem também se mostrou o melhor, pois seu resíduo absoluto médio foi o menor entre os três modelos. Isso significa que com base nos dados deste link, o número de pessoas obtido com o modelo de segunda ordem é o que mais se aproxima do número real de pessoas.

O resíduo absoluto para cada número de pessoas do modelo de segunda ordem foi obtido e pode ser visto na Tabela 20.

Tabela 20 - Resíduo absoluto - Link: Sensor 3 - Sensor 4.

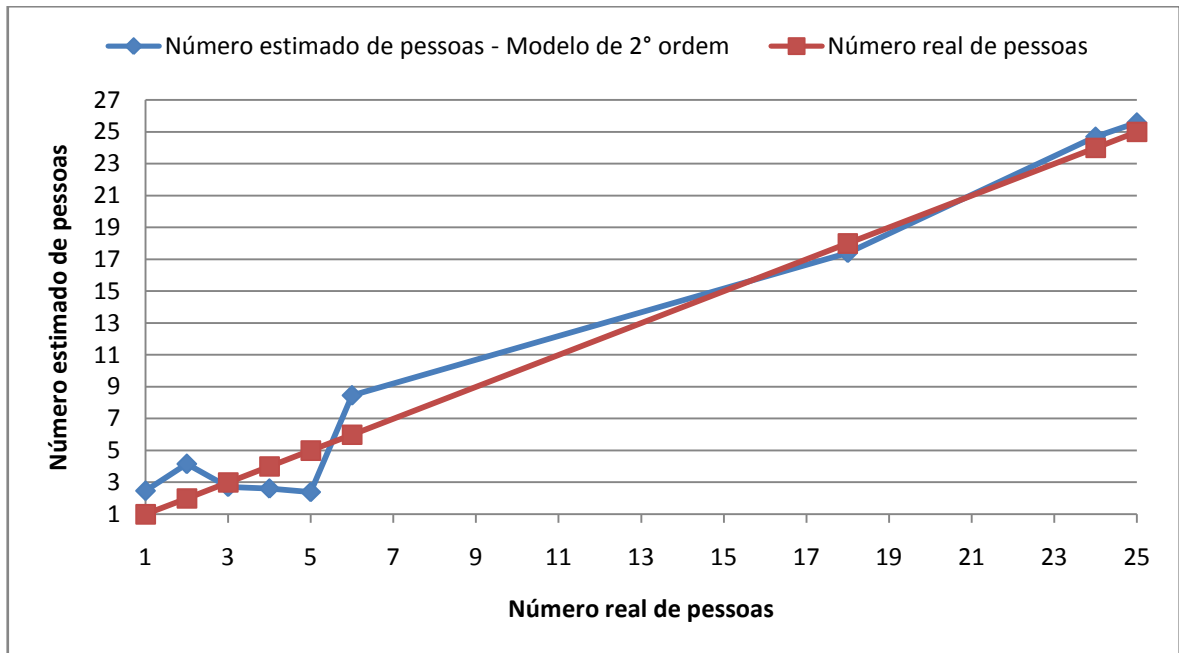
Número real de pessoas	Resíduo Absoluto
1	1.471279
2	2.1559016
3	0.2817231
4	1.3775933
5	2.5961534
6	2.4549511
18	0.5995987
24	0.6969991
25	0.572256

O grupo de pessoas que possui um resíduo absoluto maior na Tabela 20 é o de 2 pessoas com um RA maior que 2 pessoas. Este resíduo significa que o valor estimado foi muito ruim com relação ao número real de pessoas. Dois outros pontos obtiveram um resíduo maior que 2, os pontos de 5 e 6 pessoas, porém estes resíduos significam um erro de aproximadamente 50% e 35%.

Com um grande número de pessoas, como observado nos pontos 18, 24 e 25, o RA diminuiu bastante, com menos de 1 pessoa de erro.

No gráfico da Figura 21 é possível observar estes resíduos entre o número de pessoas estimadas pelo modelo de segunda ordem e o número real de pessoas.

Figura 21 - Comparação entre número real de pessoas e o número estimado de pessoas utilizando o modelo de 2° ordem para o link sensor 3-sensor 4.



É possível verificar na Figura 21 que o número estimado de pessoas e o número real de pessoas estão próximos apesar de alguns pontos demonstrando um resíduo absoluto maior do que o resíduo absoluto médio.

O modelo escolhido, com base no menor RAM foi o de segunda ordem e pode ser visto na Equação 20 após a substituição dos parâmetros estimados $\Theta(\text{média})$, $\Theta(\text{média}2)$, $\Theta(\text{desvio})$ e $\Theta(\text{variância})$.

$$Y_{\text{est}2} = X_1 * 36.876001 + X_1^2 * 45.32496 + X_2 * 47.748361 + X_2^2 * (- 31.743161) - 35.757904 \quad (20)$$

Onde:

$Y_{\text{est}2}$: é o número estimado de pessoas do modelo de 2° ordem.

X_1 : RSSI média.

X_2 : é o desvio padrão.

X_2^2 : é a variância.

5.2.5. Link: Média Geral dos Sensores

Após a análise individual de cada link, foi calculada a média geral de todos os links e calculado igualmente seu desvio padrão, variância e média quadrática.

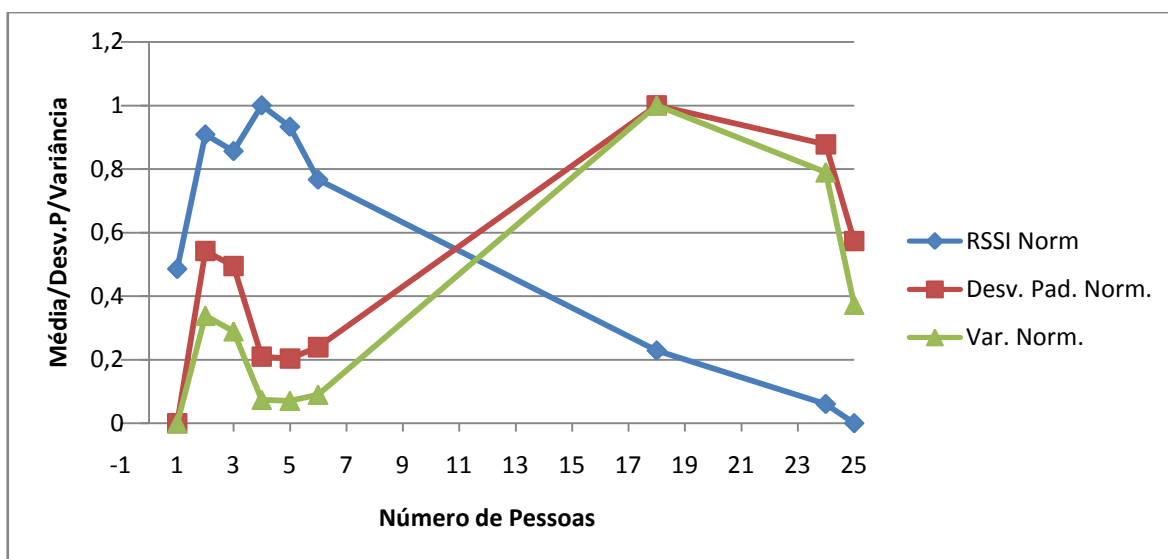
Segue a Tabela 21 com os valores calculados para a média geral dos links.

Tabela 21 - Valores de entrada calculados - Média geral dos links.

Número de Pessoas	RSSI média	Desvio Padrão	Variância	RSSI média quadrática
1	0,484897	0	0	0,498205657
2	0,90827	0,541851	0,338131	0,086093362
3	0,856453	0,494304	0,289173	0,138783186
4	1	0,209686	0,073693	0
5	0,933086	0,203406	0,070437	0,062689863
6	0,766673	0,239374	0,089958	0,221225409
18	0,228536	1	1	0,7595369
24	0,060326	0,877481	0,789256	0,935838716
25	0	0,573224	0,372467	1

Com os valores da Tabela 21 foi construído o gráfico da Figura 22 com o comportamento da RSSI média, desvio padrão e variância dos dados de modelagem.

Figura 22 - Média, Desvio Padrão e Variância - Média Geral



No gráfico da Figura 22 é observada a mesma tendência de queda da média e aumento do desvio padrão e variância de acordo com o aumento do número de pessoas.

A média geral demonstra um comportamento semelhante ao dos links 23 e 34.

Dois pontos foram considerados como *outliers*, o Ponto 1 que se mostra discrepante na média com relação aos outros pontos e o Ponto 25 referente ao desvio padrão e variância.

Após a retirada dos *outliers* os dados de modelagem foram utilizados para a estimação de parâmetros dos três modelos propostos, utilizando a técnica do MMQ. Para tal estimação foi utilizado um script no Scilab para cada um dos modelos.

Na Tabela 22 são apresentados os valores estimados dos parâmetros para cada modelo proposto.

Tabela 22 - Parâmetros estimados para cada um dos modelos propostos – Média Geral.

Parâmetro	1° ordem	2° ordem	3° ordem
$\Theta(\text{média})$	- 27.016906	266.47256	226.96737
$\Theta(\text{variância})$	- 5.1774482	18.104163	- 4.5890237
$\Theta(\text{constante})$	29.157723	- 258.55232	- 223.66978
$\Theta(\text{média}^2)$		292.14402	254.50097
$\Theta(\text{desvioP.})$		- 24.270506	

Com a obtenção dos parâmetros de cada modelo foi possível substituí-los nas equações de cada modelo proposto no item 3.1.9 para a estimação do número de pessoas.

Agora, os dados utilizados como entrada (média, desvio padrão, variância) são os de validação (índices ímpares) e os mesmos foram substituídos nas variáveis X_1 , X_2 e X_2^2 respectivamente.

Outra rotina no software Scilab foi utilizada para a estimação do número de pessoas. Após o programa ser executado, foi construída a Tabela 23 que contém em uma coluna o número real de pessoas que estavam no ambiente e o número estimado de pessoas para cada um dos três modelos propostos.

Tabela 23 - Número estimado de pessoas em cada modelo – Média Geral.

Número real de pessoas	1° ordem	2° ordem	3° ordem
2	2.7092108	1.4900242	2.6932144
3	4.4333015	2.3837834	3.7471751
4	1.7913492	4.3759564	2.9878398
5	3.5204945	4.5839628	3.6961685
6	7.9183433	5.8712499	6.1641941
18	18.103226	18.469157	17.315355
24	23.57598	23.680607	24.636764

Dada a estimação do número de pessoas foi calculado o resíduo absoluto médio (RAM) para a validação de cada modelo. Os resultados podem ser vistos na Tabela 24.

Tabela 24 - Resíduo Absoluto Médio para cada modelo – Média Geral.

	1 ordem	2 ordem	3 ordem
RAM	1.1823226	0.4050695	0.7488549

De acordo com o RAM o modelo de segunda ordem se mostrou o melhor modelo para a estimação de pessoas quando os dados da média geral dos links foram utilizados.

Este modelo retorna um erro de aproximadamente meia pessoa na estimação de pessoas.

O resíduo absoluto entre o número real de pessoas e o número estimado de pessoas pelo modelo de segunda ordem pode ser visto na Tabela 25.

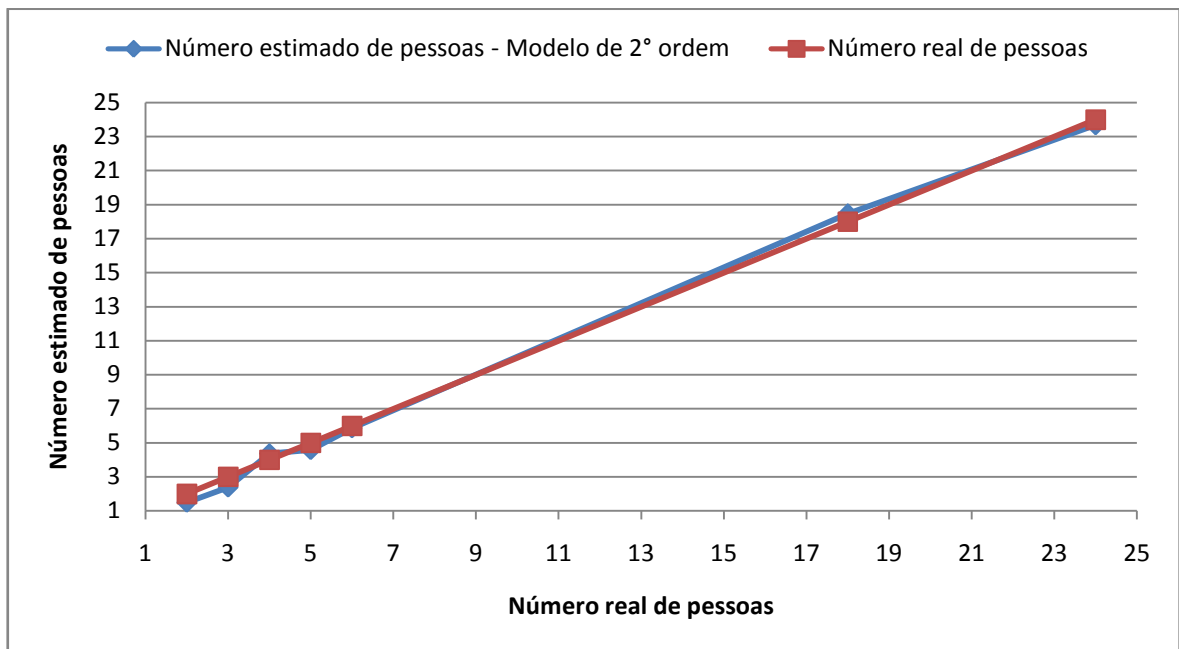
Tabela 25 - Resíduo Absoluto para o modelo de 2° ordem - Média Geral dos links.

Número real de pessoas	Resíduo Absoluto
2	0.5099828
3	0.6162236
4	0.3759494
5	0.4160442
6	0.1287571
18	0.4691505
24	0.3193999

A partir da Tabela 25, observa-se que o ponto em que se obteve o maior resíduo absoluto utilizando o modelo de segunda ordem foi para três pessoas, com erro um pouco maior que meia pessoa.

Para uma melhor observação a Figura 23 apresenta uma comparação gráfica entre o número real de pessoas e o número estimado de pessoas para o modelo de segunda ordem.

Figura 23 - Comparação entre número real de pessoas e o número estimado de pessoas utilizando o modelo de 2° ordem – Média geral dos links.



No gráfico da Figura 23 é possível verificar que a reta do número de pessoas estimadas utilizando o modelo de segunda ordem praticamente sobrepõe a reta que indica o número real de pessoas.

Portanto substituindo os valores encontrados dos parâmetros $\Theta(\text{média})$, $\Theta(\text{média}^2)$, $\Theta(\text{desvio})$ e $\Theta(\text{variância})$ do modelo de segunda ordem na equação deste mesmo modelo se chegou ao modelo que melhor estima o número de pessoas em um ambiente fechado, visto na Equação 21.

$$Y_{\text{est}2} = X_1 * 266.47256 + X_1^2 * 292.14402 + X_2 * (-24.270506) + X_2^2 * 18.104163 - 258.55232 \quad (21)$$

Onde:

$Y_{\text{est } 2}$: é o número estimado de pessoas do modelo de 2º ordem.

X_1 : RSSI média.

X_2 : é o desvio padrão.

X_2^2 : é a variância.

6. COMPARAÇÃO ENTRE *MULTILINK* E *SINGLE-LINK*

Como mencionado anteriormente, em BRAGA (2014) foi utilizado somente um link (*single-link*) para a estimação do número de pessoas em um ambiente fechado.

Foram propostos três modelos matemáticos que seguem nas equações 21, 22 e 23.

$$1^{\circ} \text{ Ordem: } Y = X_1 * K + X_2^2 * L + M \quad (21)$$

$$2^{\circ} \text{ Ordem: } Y = X_1 * K + (X_1)^2 * N + X_2 * L + (X_2)^2 * O + M \quad (22)$$

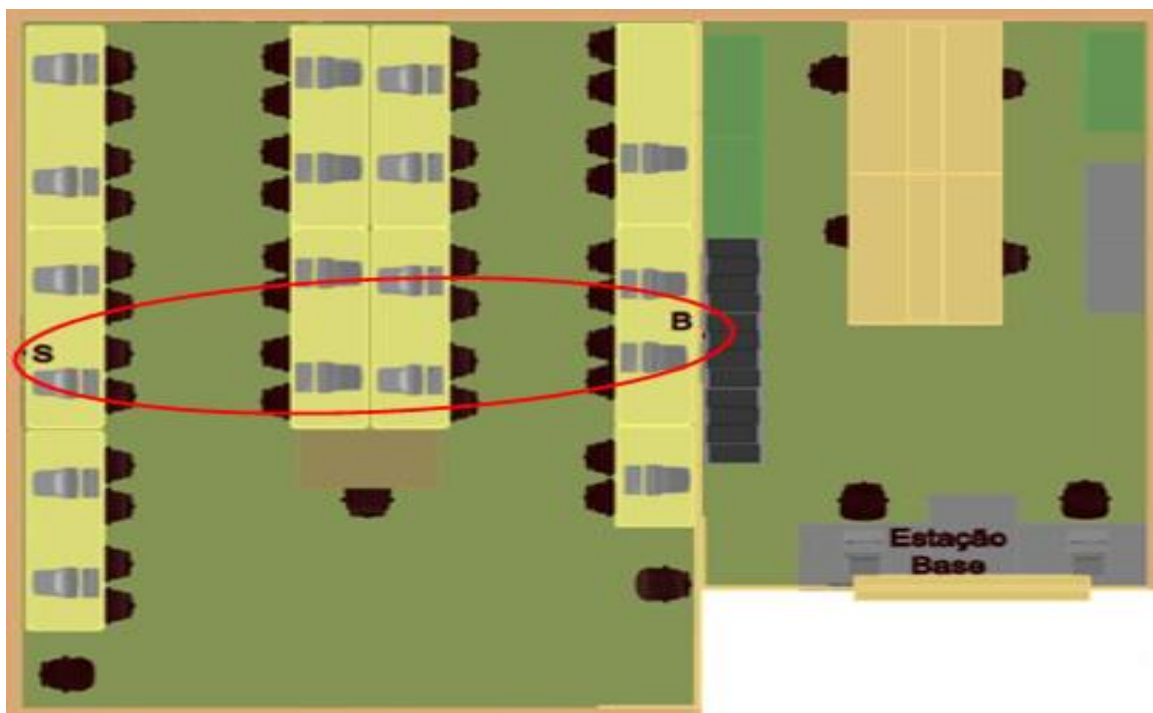
$$3^{\circ} \text{ Ordem: } Y = X_1 * K + (X_1)^2 * N + (X_1)^3 * P + X_2 * L + (X_2)^2 * O + (X_2)^3 * Q + M \quad (23)$$

Onde Y é o número estimado de pessoas, X_1 é a média da RSSI, X_2 é a variância e K, L, M, N, O, P, Q são os parâmetros a serem estimados.

Como dados de entrada nos modelos foram utilizados a média e a variância.

Foi utilizado um link (*single-link*) para a coleta de dados no ambiente de testes. A Figura 24, retirada de BRAGA (2014), ilustra a disposição da base e do sensor que compõe o link.

Figura 24 - Vista de cima do ambiente de testes efetuados por (BRAGA, DANIEL B.) - Disposição do link.



Na Figura 24, é possível observar que o link ficou posicionado praticamente no meio do laboratório. A base é representada pela letra “B” e o Sensor pela letra “S”. Os nós sensores foram posicionados a uma altura de 1,3 metros.

Tanto na base como no sensor foi utilizado um dispositivo RF-Bee que é similar aos dispositivos utilizados para coleta de dados deste trabalho.

A potência utilizada foi a de 10 mW, que em dBm é o valor de 10 dBm e a frequência de operação utilizada foi a de 915 MHz, a mesma utilizada neste trabalho.

Foi utilizado o mesmo método para estimação dos parâmetros e estimação da taxa de ocupação que este trabalho, o MMQ.

Após a remoção de *outliers*, estimação de parâmetros e substituição dos parâmetros e dados de validação (média e variância), (BRAGA, 2014) chegou aos valores indicados na Tabela 26 para a estimação de pessoas.

Tabela 26 - Número estimado de pessoas utilizando *single-link* em BRAGA (2014).

Número Real de Pessoas	Número Estimado de Pessoas		
	1° Ordem	2° Ordem	3° Ordem
2	1,42	2,1	12,71
7	11,76	10,29	10,49
11	13,23	12,29	10,5
12	11,11	9,21	8,47
18	17,07	19,27	18,4
22	22,75	24,95	27,9
27	33,09	30,53	31,57
30	31,16	29,01	43,94
32	33,09	30,53	31,57

O resíduo absoluto médio encontrado para cada um dos três modelos pode ser visto na Tabela 27.

Tabela 27 - Resíduo absoluto médio encontrado em cada modelo proposto em BRAGA (2014).

	1° Ordem	2° Ordem	3° Ordem
Resíduo Absoluto Médio	2,053245	1,964668	4,829152

Portanto, o modelo proposto em BRAGA (2014) foi o modelo de segunda ordem com um RAM um pouco abaixo de 2 pessoas.

Fazendo uma comparação entre BRAGA (2014) e este trabalho, no modelo de primeira ordem foram utilizados exatamente os mesmos dados de entrada.

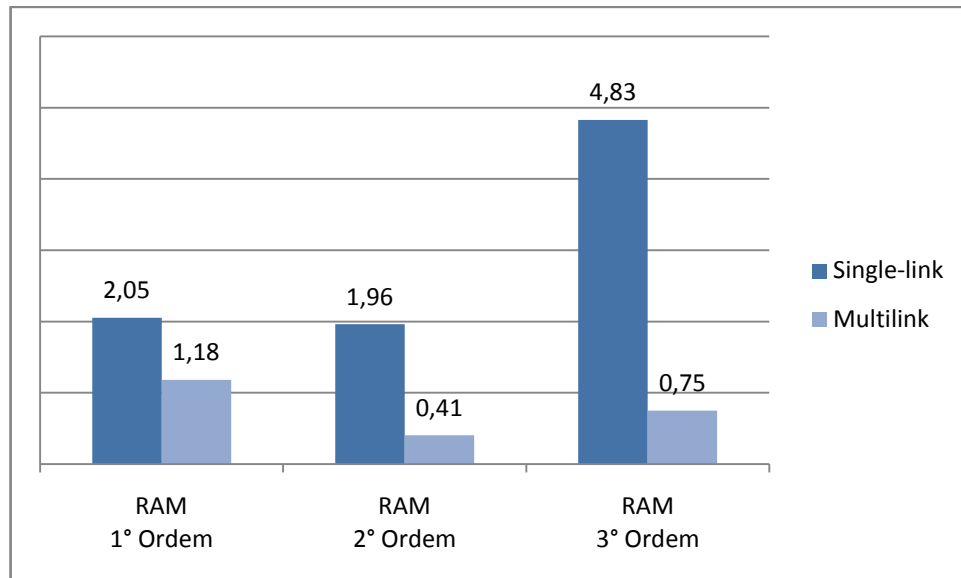
No modelo de segunda ordem foram utilizados neste trabalho, além da variância e média da RSSI utilizados em BRAGA (2014) foi utilizado o desvio padrão.

O modelo de terceira ordem em BRAGA (2014) se mostrou com um RAM muito grande comparando-se com os outros dois modelos propostos e neste trabalho, com base neste resultado este trabalho propôs um novo modelo de terceira ordem e que pode ser visto na Equação 9 (Capítulo 3 – item 3.1.9).

Com base nas diferenças mencionadas acima, além da utilização de *multilink* neste trabalho, foi elaborado um gráfico de colunas comparando o RAM

de cada um dos três modelos de BRAGA (2014) e o RAM encontrado como resultado deste trabalho e pode ser visto na Figura 25.

Figura 25 - Comparação do resíduo absoluto médio encontrado em (BRAGA, DANIEL B.) com single-link e encontrado neste trabalho com multilink.



Com base na Figura 25 é importante fazer as seguintes observações:

- Para o modelo de primeira ordem, com a utilização do *multilink* para a estimação de pessoas houve uma melhora de aproximadamente 42% com relação ao *single-link*. Como já falado este modelo foi utilizado de maneira idêntica em ambos os trabalhos, ou seja, comparando somente a utilização de *single-link* e *multilink* com os mesmos dados de entrada, o segundo método se mostrou mais preciso que o primeiro.

- Para o modelo de segunda ordem, houve uma melhora de aproximadamente 79%. Esta melhora pode ser creditada à utilização do *multilink* e também com a utilização do desvio padrão e variância ao invés de utilizar a variância e variância quadrática como foi feito em BRAGA (2014).

- Já para o modelo de terceira ordem a melhora na estimação de pessoas, de acordo com o RAM, foi de aproximadamente 84%. Portanto pode-se considerar que este modelo proposto neste trabalho também pode ser utilizado

para a estimação de pessoas pelo fato de que a diferença de RAM entre o modelo de segunda ordem e o de terceira ordem foi de aproximadamente 0,34 enquanto a diferença de RAM dos mesmos modelos em BRAGA (2014) foi de quase 3 pessoas.

Vale a pena destacar que todos os três modelos propostos neste trabalho obtiveram um resíduo menor que o modelo proposto em BRAGA (2014).

7. CONCLUSÃO

Foi instalada uma rede de sensores sem fio em 915 MHz em um ambiente interno (laboratório de Meios de Transmissão da Puc-Campinas) para coletar dados, RSSI de cada link, para a obtenção de um modelo matemático capaz de determinar a taxa de ocupação do ambiente.

Para se chegar ao modelo matemático, primeiramente foi feito o acompanhamento de aulas neste laboratório a fim de poder separar os grupos de pessoas (número de pessoas em cada intervalo de tempo) para utilização posterior dos dados coletados na estimação de pessoas.

Três modelos foram propostos, de primeira, segunda e terceira ordem.

Uma base e quatro nós sensores foram utilizados somando 4 links independentes.

Primeiramente foi feita análise de cada link de forma individual e por último foi feita a análise da média geral dos links, que é a proposta do trabalho.

Portanto a proposta deste trabalho foi determinar a taxa de ocupação utilizando uma RSSF com múltiplos links e utilizando a média geral da RSSI de todos os links tendo como dados de entrada a média da RSSI, o desvio padrão e a variância. Um modelo híbrido de terceira ordem foi proposto com base no modelo de terceira ordem encontrado em BRAGA (2014) que, utilizando um único link (*single-link*) chegou a um erro muito maior se comparado com o modelo final proposto por ele.

O resultado encontrado com a utilização de múltiplos links para a determinação da taxa de ocupação mostrou que é possível estimar o número de pessoas utilizando um único link, porém conforme mostrado no Capítulo 5 de resultados os mesmos variam de um para outro.

Essa variação possivelmente se deve à movimentação das pessoas no ambiente e a disposição que elas ficam no mesmo. Como são múltiplos links, uma pessoa pode ficar diretamente obstruindo um link enquanto o outro está sem nenhuma obstrução.

Com a utilização da média geral dos links, apesar de um link se mostrar com uma grande variação da RSSI e desvio padrão e obter um RAM maior comparando com outros links melhores (como por exemplo, o link B1 se

comparado com o link 23), a média geral dos links mostra que um link compensa o outro e é possível chegar à determinação da taxa de ocupação com um RAM baixo. Portanto é possível afirmar que múltiplos links, tratados de forma única, no caso a média geral dos links, melhoram, pois agregam informação distribuída.

Em um link individual o RAM encontrado foi de aproximadamente 4,45 utilizando o modelo de primeira ordem enquanto na média geral dos links o menor RAM encontrado foi utilizando o modelo de segunda ordem, com aproximadamente 0,4.

O modelo proposto em BRAGA (2014), que utiliza *single-link* foi o de segunda ordem, tendo como dados de entrada RSSI média e variância e obteve um RAM de aproximadamente 2 pessoas.

A utilização do *multilink*, proposto neste trabalho, com a média geral dos links obteve um RAM de aproximadamente 0,4 pessoas também utilizando o modelo de segunda ordem. Estes dados mostram uma melhora de aproximadamente 79% com a utilização do *multilink*.

Neste trabalho os sensores foram posicionados de forma a obter o link das duas diagonais do ambiente. Como trabalhos futuros uma possível melhoria seria o estudo do comportamento dos alunos no laboratório de forma a ter o melhor posicionamento dos sensores. Neste caso, como o modelador teria certo conhecimento do comportamento do ambiente, poderia ser considerado modelagem caixa cinza.

Outros trabalhos interessantes como sugestão seria a implementação adaptativa em tempo real e a estimação ponderada das médias para o multilink, dando pesos para cada link.

Outra mudança que poderia ser interessante é a realização dos mesmos testes utilizando RSSF em 2.4GHz.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] HOLGER, KARL; WILLING, ANDREAS. “*Protocols and Architectures for Wireless Sensor Networks.*”, First Edition, Wiley, 2005.
- [2] ILYAS, MOHAMMAD; MAHGOUB, IMAD. “*Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems.*”, CRC Press, 2004.
- [3] ABUSSANDRA, J.M., “*O Edifício Residencial Paulistano nos Anos 70, 80 e 90: A Arquitetura e o Mercado Imobiliário*”. 2007. 254 f. Dissertação (Faculdade de Arquitetura e Urbanismo Curso de Pós – Graduação) – Universidade São Judas Tadeu. 2007.
- [4] STROPARO, JOELSON RICARDO et al. “*Estudo da Taxa de Ocupação do Centro Cirúrgico Através da Modelagem e Simulação de Sistemas*”. Programa de Pós Graduação em Tecnologia em Saúde, PUC-Paraná, 2004.
- [5] SWANK, ERIC; CLAPP, JOHN D. “*Some Methodical Concerns When Estimating the Size of Organizing Activities*”, Journal of Community Practice, Vol.3, 1999.
- [6] RAPPAPORT, T.S. “*Wireless Communications – Principle and Practice*”, 2nd edition, Prentice Hall.
- [7] MOTA, L.T.M., “*Métodos de Previsão do Comportamento da Carga na Recomposição de Sistemas de Energia Elétrica*”. 2005. 183f. Tese (Doutorado em Engenharia Elétrica) – Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, SP.
- [8] COTA, FELIPE E.M. et al, “*Modelagem Matemática dos Sistemas de Temperatura da Planta Didática de Automação Industrial SMAR-PD3*”, Departamento de Engenharia Elétrica, Universidade Federal de Viçosa, 2010.
- [9] PROCÓPIO, EDSON TAIRA, “*Identificação de modelo matemático para representar a transmissão de rede de sensores sem fio em ambientes construídos.*” 2012. 159f. Dissertação (Mestrado Profissional em Gestão de Redes de Telecomunicações) – Pontifícia Universidade Católica de Campinas, São Paulo.
- [10] BRANDOLT, HENRY G., “*Simulação de escoamento em dutos por caracterização de eventos*”, Dissertação (Mestrado em Engenharia Química) – Universidade Federal de Santa Catarina, 2002.

- [11] BRAGA, DANIEL B., “*Determinação da Taxa de Ocupação de Ambientes Internos Fechados em Função da Potência de Sinal Recebido em Redes de Sensores sem Fio.*”Dissertação (Mestrado Profissional em de Redes de Telecomunicações) – Pontifícia Universidade Católica de Campinas, 2014.
- [12] RADIOIT. “*Datasheet BE900*”. Disponível em www.radioit.com.br, acessado em 20 de Outubro de 2014.
- [13] SITE UOL; “*Protestos contra governo reúnem 2 milhões pelo Brasil, dizem PM’s*”. Disponível em <http://noticias.uol.com.br/politica/ultimas-noticias/2015/03/15/protestos-contra-governo-e-corrupcao-reunem-mais-de-2-milhoes-pelo-brasil-dizem-pms.htm>, acessado em 12 de Maio de 2015.
- [14] AGUIRRE, L.A.; RODRIGUES, G.G.; JÁCOME, C.R.F. “*Identificação de Sistemas Não Lineares Utilizando Modelos Narmax Polinomiais: Uma Revisão e Novos Resultados*”. SBA Controle e Automação, Campinas, V. 9, n.2, p. 90-106 mai/ago. 1998. Disponível em: < <http://www.sba.org.br/revista/vol9/V9p90.pdf>>. Acessado em 03 de Abril de 2015.
- [15] RADIUINO. Disponível em <http://www.loja.radiuino.com.br/products/>, acessado em 20-03-2015 às 20h30min.
- [16] AKHLAGHINIA, JAVAD M.; “*Occupancy Monitoring in Intelligent Environment Trough Integrated Wireless Localizing Agents*”. IEEE Symposium, pgs 70-76, 2009.
- [17] SUBBURAMAN V.B., DESCAMPS A. e CARINCOTTE C., “*Couting People in the Crowd Using a Generic Head Detector*”, Image Departament, Multitel absI, 2012.

ANEXO A – Software de coleta de Dados Python

```
# PROGRAMA PARA MEDIDAS NO LABORATÓRIO DE MEIOS DE
TRANSMISSÃO - CORRENTE COM 4 NÓS SENSORES

import serial
import math
import time
import struct
from time import localtime, strftime

n_serial = 11
# Configura a serial
# para COM# o número que se coloca n-1 no primeiro parâmetro. Ex
COM9 valor 8
#n_serial = raw_input("Digite o número da serial = ") #seta a serial
n_serial1 = int(n_serial) - 1
ser = serial.Serial(n_serial1, 9600, timeout=0.5,parity=serial.PARITY_NONE) #
seta valores da serial

# Identificação da base
#ID_base = raw_input("ID_base = ")
ID_base = 0
ID_primeiro_sensor = 1
Numero_saltos = 4
Pacote_descida = 1

Contador = 0

# Cria o vetor Pacote
Pacote = {}
j=0
```

```
RSSIdB1=RSSId12= RSSId23= RSSId34= RSSlu43= RSSlu32= RSSlu21=
RSSlu1B=0
```

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

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

```
    Pacote[i] = 0
```

```
while True:
```

```
    try:
```

```
        # Imprime na tela o menu de
```

```
        #print 'Escolha um comandos abaixo e depois enter'
```

```
        print '1 - Realiza medidas:'
```

```
        print 's - Para sair:'
```

```
        Opcao = raw_input('Entre com a Opção = ')
```

```
        Opcao = 1
```

```
        # Limpa o buffer da serial
```

```
        ser.flushInput()
```

```
        # VÃO EXISTIR A BASE - NÓ SENSOR 1 - NÓ SENSOR 2 - NÓ SENSOR 3 -
        NÓ SENSOR 4
```

```
        Pacote[8] = int(ID_primeiro_sensor) # Endereço do sensor NetHdr[0] que é o
        byte 8
```

```
        Pacote[9] = int(Pacote_descida) # variável que diz que é pacote de descida
        TranspHdr[3] byte 15
```

```
        Pacote[10] = int(ID_base) # NetHdr[2] byte 10
```

```
        Pacote[11] = int(Numero_saltos) # número de saltos da rede NetHdr[3] byte 11
```

```
        # Leitura de temperatura e luminosidade
```

```
        if Opcao == int(Opcao):
```

```
            num_medidas = raw_input('NUMERO DE MEDIDAS = ')
```

```

w = int(num_medidas)+1

filename = raw_input(' NOME DO ARQUIVO = ')
filename1 = filename
print "Arquivo de log: %s" % filename1
S = open(filename1, 'w')

#=====For para realização das medidas

for j in range(1,w):

    # TRANSMISSÃO DO PACOTE
    for k in range(0,52): # transmite pacote
        TXbyte = chr(Pacote[k])
        ser.write(TXbyte)

        # Aguarda a transmissão para o Nó 1 - Nó 2 - Nó 3 - Nó 4 - Nó 3 - Nó 2 -
Nó 1 - Base
        time.sleep(0.3)

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

            Contador = Contador + 1

#===== RSSIdB1
            rssidB1 = ord(line[16]) # RSSI_DownLink

            if rssidB1 > 128:
                RSSIdB1=((rssidB1-256)/2.0)-74
            else:
                RSSIdB1=(rssidB1/2.0)-74

```

```

#===== RSSId12
    rssid12 = ord(line[17]) # RSSI_DownLink

    if rssid12 > 128:
        RSSId12=((rssid12-256)/2.0)-74
    else:
        RSSId12=(rssid12/2.0)-74

#===== RSSId23
    rssid23 = ord(line[18]) # RSSI_DownLink

    if rssid23 > 128:
        RSSId23=((rssid23-256)/2.0)-74
    else:
        RSSId23=(rssid23/2.0)-74

#===== RSSId34
    rssid34 = ord(line[19]) # RSSI_DownLink

    if rssid34 > 128:
        RSSId34=((rssid34-256)/2.0)-74
    else:
        RSSId34=(rssid34/2.0)-74

#===== RSSlu43
    rssi43 = ord(line[20]) # RSSI_DownLink

    if rssi43 > 128:
        RSSlu43=((rssi43-256)/2.0)-74
    else:
        RSSlu43=(rssi43/2.0)-74

```

```

#===== RSSlu32
    rssi32 = ord(line[21]) # RSSI_DownLink

    if rssi32 > 128:
        RSSlu32=((rssi32-256)/2.0)-74
    else:
        RSSlu32=(rssi32/2.0)-74

#===== RSSlu21
    rssi21 = ord(line[22]) # RSSI_DownLink

    if rssi21 > 128:
        RSSlu21=((rssi21-256)/2.0)-74
    else:
        RSSlu21=(rssi21/2.0)-74

#===== RSSlu1B
    rssi1B = ord(line[2]) # RSSI_DownLink

    if rssi1B > 128:
        RSSlu1B=((rssi1B-256)/2.0)-74
    else:
        RSSlu1B=(rssi1B/2.0)-74

#===== Escreve na tela e no arquivo
    print Contador,' RSSIdB1 = ',RSSIdB1,' RSSId12 = ',RSSId12,
    'RSSId23 = ', RSSId23 , 'RSSId34 = ' ,RSSId34, 'RSSlu43', RSSlu43, 'RSSlu32',
    RSSlu32, ' RSSlu21 = ', RSSlu21 , 'RSSlu1B = ',RSSlu1B
    print >>S,time.asctime(),Contador,'RSSIdB1 = ',RSSIdB1,'RSSId12 =
    ',RSSId12, 'RSSId23 = ', RSSId23, 'RSSId34 = ', RSSId34, 'RSSlu43 = ',
    RSSlu43, 'RSSlu32 = ', RSSlu32 ,'RSSlu21 = ',RSSlu21,'RSSlu1B = ',RSSlu1B

    else:

```

```
print '-----'  
ser.flushInput()  
  
#time.sleep(0.3)  
  
S.close()  
  
if Opcao == "s" or Opcao == "S":# caso o caracter digitado for s  
    ser.close() # fecha a porta COM  
    print 'Fim da ExecuÃ§Ã£o' # escreve na tela  
  
    break  
  
except KeyboardInterrupt:  
    ser.close()  
    break
```

ANEXO B – Protocolo de Roteamento Fixo de 4 saltos

Firmware da Base:

// NET : classe da camada de Rede

// Mais informações em www.radiuino.cc

// Copyright (c) 2011

// Author: Pedro Henrique Gomes e Omar C. Branquinho

// Versão 1.0: 12/09/2011

// Este arquivo é parte da plataforma Radiuino

// Este programa é um software livre; você pode redistribuí-lo e/ou modificá-lo dentro dos termos da Licença Pública Geral Menor GNU

// como publicada pela Fundação do Software Livre (FSF); na versão 2 da Licença, ou (na sua opinião) qualquer futura versão.

// Este programa é distribuído na esperança que possa ser útil, mas SEM NENHUMA GARANTIA; sem uma garantia implícita

// de ADEQUAÇÃO a qualquer MERCADO ou APLICAÇÃO EM PARTICULAR. Veja a Licença Pública Geral Menor GNU para maiores detalhes.

// Você deve ter recebido uma cópia da Licença Pública Geral Menor GNU junto com este programa, se não, escreva para a Fundação

// do Software Livre(FSF) Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

// This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License

// as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. This library

// is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY


```
// or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General
Public License for more details. You should have received a copy
// of the GNU Lesser General Public License along with this library; if not, write to
the Free Software Foundation, Inc., 51 Franklin St,
// Fifth Floor, Boston, MA 02110-1301 USA
```

```
#include "Headers.h"
```

```
/**
```

```
 * Construtor da camada de Rede.
```

```
 */
```

```
NET::NET()
```

```
{
```

```
    my_addr = 0; /* Endereço */
```

```
}
```

```
/**
```

```
 * Inicializa a camada de Controle de Acesso ao Meio.
```

```
 */
```

```
void NET::initialize(void)
```

```
{
```

```
}
```

```
/**
```

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

```
 */
```

```
inline void NET::send(packet * pkt)
```

```
{
```

```
    return;
```

```
}
```

```
/**
```

```
 * Recebe o pacote da camada inferior
```

```

*/
inline void NET::receive(packet * pkt)
{
    return;
}

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

```

Firmware dos nós sensores:

```

// NET : classe da camada de Rede

// Mais informações em www.radiuino.cc
// Copyright (c) 2011
// Author: Pedro Henrique Gomes e Omar C. Branquinho
// Versão 1.0: 12/09/2011

// Este arquivo é parte da plataforma Radiuino
// Este programa é um software livre; você pode redistribuí-lo e/ou modificá-lo
dentro dos termos da Licença Pública Geral Menor GNU
// como publicada pela Fundação do Software Livre (FSF); na versão 2 da
Licença, ou (na sua opinião) qualquer futura versão.
// Este programa é distribuído na esperança que possa ser útil, mas SEM
NENHUMA GARANTIA; sem uma garantia implícita
// de ADEQUAÇÃO a qualquer MERCADO ou APLICAÇÃO EM PARTICULAR.
Veja a Licença Pública Geral Menor GNU para maiores detalhes.
// Você deve ter recebido uma cópia da Licença Pública Geral Menor GNU junto
com este programa, se não, escreva para a Fundação
// do Software Livre(FSF) Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301
USA

```

```
// This library is free software; you can redistribute it and/or modify it under the
// terms of the GNU Lesser General Public License
// as published by the Free Software Foundation; either version 2 of the License,
// or (at your option) any later version. This library
// is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY;
// without even the implied warranty of MERCHANTABILITY
// or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General
// Public License for more details. You should have received a copy
// of the GNU Lesser General Public License along with this library; if not, write to
// the Free Software Foundation, Inc., 51 Franklin St,
// Fifth Floor, Boston, MA 02110-1301 USA
```

```
#include "Headers.h"
```

```
/**
```

```
 * Construtor da camada de Rede.
```

```
 */
```

```
NET::NET()
```

```
{
```

```
//+++++
```

```
+++++
```

```
//          ENDEREÇO DO SENSOR
```

```
//+++++
```

```
+++++
```

```
    my_addr = 1; /* Endereço */
```

```
    my_dest = 0; // destino
```

```
    my_neighbor = 2; //vizinho
```

```
}
```

```
/**
```

```
 * Inicializa a camada de Controle de Acesso ao Meio.
```

```
 */
```

```

void NET::initialize(void)
{
}

/**
 * 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;
}

/**
 * Envia o pacote para a camada inferior
 */
inline void NET::send(packet * pkt)
{
    /* Envia para a camada inferior */
    Mac.send(pkt);

    return;
}

/**
 * Recebe o pacote da camada inferior
 */
inline void NET::receive(packet * pkt)
{
    // Foi criada uma variável index para ser um índice de qual byte são
    armazenados os valores de RSSI
    int index; //posicao para gravar os RSSI's
}

```

```

// PRIMEIRO IF – PACOTE DECENDO
// Verifica se o pacote é para o sensor, se é um pacote de descida e se ainda não
chegou no último sensor, verificando se o número de saltos é >3

if((pkt->NetHdr[0]==my_addr) && (pkt->NetHdr[1] == 1) && (pkt->NetHdr[3]>1))
//Se quantidade de saltos na rede > que 1 e se pkt esta marcado como pacote de
roteamento RSSI
{
    // Coloca no pacote o endereço do nó vizinho para onde vai o pacote e
o seu endereço para dizer quem enviou
    pkt->NetHdr[0] = Net.my_neighbor; //coloco o endereço como destino
(descida) do nó e no byte 8 que é o NetHdr[0]
    pkt->NetHdr[2] = Net.my_addr; // coloca o próprio endereço como
origem no byte 10 que é NetHdr[2]

    // Decrementa o número de saltos pois passou por este sensor
    pkt->NetHdr[3]=pkt->NetHdr[3]-1; //decremento o numero de saltos

    // Código para armazenamento de RSSI
    index=pkt->Data[35]; // recebe o que está na posição 35. Como é o
primeiro vai estar zero o byte incrementa o índice para armazenamento no byte
correto a RSSI no vetor Data[]. Ou seja, o próximo sensor já sabe onde gravar,
em função da linha acima
    pkt->Data[35]=pkt->Data[35]+1; // armazeno o RSSI de down na
posicao correta e incremento posicao para o proximo sensor
    pkt->Data[index]=pkt->PhyHdr[0]; // como o índice já está certo é
possível gravar a RSSI no local correto

    // Envia pacote para a camada MAC
    Mac.send(pkt); //despacho o pacote pro meu vizinho
    return;
}

```

```

// SEGUNDO IF – AÇÃO DO ÚLTIMO SENSOR DA REDE
// if se for o último sensor
if((pkt->NetHdr[0]==my_addr) && (pkt->NetHdr[1]==1) && (pkt->NetHdr[3]==1))
//se quant. de saltos for = 1, então sou o último sensor da contagem
{
    // observar que a partir de agora o pacote vai subir
    pkt->NetHdr[0] = Net.my_dest; // esse é o endereço do destino mesmo,
    pois vai ter que subir e não o endereço de vizinho
    pkt->NetHdr[2] = Net.my_addr; // coloca o endereço

    // INDICAÇÃO QUE O PACOTE AGORA TEM QUE SUBIR
    pkt->NetHdr[1]=2; //importante: marco o pacote para retornar obtendo os
    RSSI's de UP

    // armazenamento da RSSI
    index=pkt->Data[35];
    pkt->Data[35]=pkt->Data[35]+1; // incrementa index
    pkt->Data[index]=pkt->PhyHdr[0]; // armazeno o RSSI de down na
    posicao correta este é o byte 0 do pacote

    // Envia pacote para a camada MAC
    Mac.send(pkt); //despacho o pacote pro meu destino
    return;
}

// TERCEIRO IF
// TEM QUE TESTAR O PACOTE QUE CHEGA DO VIZINHO. PODE SER UM
//PACOTE QUE ESTÁ VOLTANDO NA CORRENTE
// LEMBRAR QUE TEM QUE CONTINUAR GRAVANDO AS RSSIs DE SUBIDA
AGORA

```

```

if((pkt->NetHdr[0]==my_addr) && (pkt->NetHdr[1]==2) && (pkt->NetHdr[2] ==
Net.my_neighbor) ) //se quem enviou o pacote for meu vizinho,entao irei passar
para o meu destino
    {
        // Envia pacote para o nó destino, que
        pkt->NetHdr[0] = Net.my_dest; //coloco meu destino
        pkt->NetHdr[2] = Net.my_addr; //me coloco como origem

        index=pkt->Data[35]; //recebe a posicao onde gravará o RSSI
        pkt->Data[35]=pkt->Data[35]+1; //incrementa a posição para o proximo
sensor
        pkt->Data[index]=pkt->PhyHdr[0]; //armazena a RSSI de UP

        Mac.send(pkt); //despacho o pacote
        return;
    }
}

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

```

ANEXO C – Código de tratamento de dados no Scilab

- Rotina para separação dos dados em pares e ímpares e cálculo da média, desvio padrão, variância e média quadrática.

```
//Thiago Tortorelli de Faria
//Mestrado em Engenharia Elétrica
//Divisão de dados em índices pares (dados de modelagem) e índices ímpares (dados de
validação)
clc();

[arquivo] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes
Lab de Meios\mmq\Resultados\Comparacao potencias\rssi menos10dBm 13pessoa.txt");
tamanho = size (arquivo);

//z=input("digita valor do z: ");
n=round ((tamanho(2)/2));

for i = 1:n
    pares(i)=arquivo(2*i);
    ímpares(i)=arquivo(2*i+1);
end

disp (pares,"pares=");

disp (ímpares, "ímpares=");

////////INDICES PARES

disp("INDICES PARES");

//calcula a rssi de indice par em watt
rssi_par_watt=10^(pares/10);
//disp (rssi_par_watt);

//calcula a média da rssi de indice par em watt
rssi_par_watt_media = mean(rssi_par_watt);
//disp (rssi_par_watt_media, "rssi média em watt =");

//transforma a rssi media de indice par em watt para dBm
rssi_par_media_dBm=10*log10(rssi_par_watt_media);
disp (rssi_par_media_dBm, "rssi média em dBm dos índices pares =");

//Calcula o desvio padrão dos dados pares em dBm
desviop_par=stdev (pares);
disp(desviop_par, "desvio padrão dos índices pares =");

//Calcula a variância dos dados pares em dBm
var_par=desviop_par^2;
disp(var_par,"variância dos índices pares=");

//Calcula a rssi média quadrática dos índices pares (direto em dBm)
media2_par=rssi_par_media_dBm^2;
```



```

disp(media2_par,"rssi média ao quadrado dos dados de indice par=");

////////INDICES ÍMPARES

disp("INDICES ÍMPARES");

//calcula a rssi de indice ímpar em watt
rssi_ímpar_watt=10^(ímpares/10);
//disp (rssi_par_watt);

//calcula a média da rssi de indice ímpar em watt
rssi_ímpar_watt_media = mean(rssi_ímpar_watt);
//disp (rssi_ímpar_watt_media, "rssi média em watt =");

//transforma a rssi media de indice ímpar em watt para dBm
rssi_ímpar_media_dBm=10*log10(rssi_ímpar_watt_media);
disp (rssi_ímpar_media_dBm, "rssi média em dBm dos indices ímpares =");

//Calcula o desvio padrão dos dados ímpares em dBm
desviop_ímpar=stdev (ímpares);
disp(desviop_ímpar, "desvio padrão dos indices ímpares =");

//Calcula a variância dos dados ímpares em dBm
var_ímpar=desviop_ímpar^2;
disp(var_ímpar,"variancia dos índices ímpares=");

//Calcula a rssi média quadrática dos indices ímpares (direto em dBm)
media2_ímpar=rssi_ímpar_media_dBm^2;
disp(media2_ímpar,"rssi média ao quadrado dos dados de indice ímpar=");

clear

```

- Rotina para estimação dos parâmetros

Modelo de Primeira Ordem:

```
//Thiago Tortorelli de Faria
```

```

////////////////////////////////////
// Calculando o modelo 1 para a media geral dos links
////////////////////////////////////

```

```
clc(); //limpa o console antes de começar
```

```
[np] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab de Meios\mmq\Resultados\l23\sem_outliers\npessoas.txt");
[L_linhas, L_colunas] = size(np);
```

```
[media] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab de Meios\mmq\Resultados\lgeral\sem_outliers\media_geral_par_sout.txt");
[L_linhas, L_colunas] = size(media);
```

```
[var] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab de Meios\mmq\Resultados\lgeral\sem_outliers\var_geral_par_sout.txt");
[L_linhas, L_colunas] = size(var);
```

```

[dp] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab de Meios\mmq\Resultados\lgeral\sem_outliers\desvp_geral_par_sout.txt");
[L_linhas, L_colunas] = size(dp);

[one] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab de Meios\mmq\Resultados\l23\sem_outliers\one.txt");
[L_linhas, L_colunas] = size(one);

[media2] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab de Meios\mmq\Resultados\lgeral\sem_outliers\media2_geral_par_sout.txt");
[L_linhas, L_colunas] = size(media2);

y=np;
x=[media,var,one];

//Calcula teta
teta = ((x)*x)^-1*(x)*y;
disp("Teta média geral dos links:");
disp (teta);

//Calculo dos resíduos

//resíduo simples
rs = y-(x*teta);
disp (rs, "residuo simpes =");

//resíduo médio
rm= mean(rs);
disp (rm, "residuo médio =");

//resíduo absoluto médio
ram = mean(abs(rs));
disp (ram, "residuo absoluto médio =");

```

Modelo de Segunda Ordem:

```

//Thiago Tortorelli de Faria

////////////////////////////////////
// Calculando o modelo 2 para a media geral dos links
////////////////////////////////////

clc(); //limpa o console antes de começar

[np] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab de Meios\mmq\Resultados\l23\sem_outliers\npessoas.txt");
[L_linhas, L_colunas] = size(np);

[media] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab de Meios\mmq\Resultados\lgeral\sem_outliers\media_geral_ímpar_sout.txt");
[L_linhas, L_colunas] = size(media);

[var] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab de Meios\mmq\Resultados\lgeral\sem_outliers\var_geral_ímpar_sout.txt");
[L_linhas, L_colunas] = size(var);

```

```
[dp] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab de Meios\mmq\Resultados\lgeral\sem_outliers\desvp_geral_ímpar_sout.txt");
[L_linhas, L_colunas] = size(dp);
```

```
[one] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab de Meios\mmq\Resultados\l23\sem_outliers\one.txt");
[L_linhas, L_colunas] = size(one);
```

```
[media2] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab de Meios\mmq\Resultados\lgeral\sem_outliers\media2_geral_ímpar_sout.txt");
[L_linhas, L_colunas] = size(media2);
```

```
y=np;
x=[media,media2,dp,var,one];
```

```
//Calcula teta
teta = ((x')*x)^-1*(x')*y;
disp("Teta A calculado:");
disp (teta, "teta media geral modelo 2 =");
```

```
//resíduo simples
rs = y-(x*teta);
disp (rs, "residuo simples =");
```

```
//resíduo médio
rm= mean(rs);
disp (rm, "resíduo médio =");
```

```
//resíduo absoluto médio
ram = mean(abs(rs));
disp (ram, "resíduo absoluto médio =");
```

Modelo de Terceira Ordem

```
//Thiago Tortorelli de Faria
```

```
////////////////////////////////////
// Calculando o modelo 3 para a média geral
////////////////////////////////////
```

```
clc(); //limpa o console antes de começar
```

```
[np] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab de Meios\mmq\Resultados\l23\sem_outliers\npessoas.txt");
[L_linhas, L_colunas] = size(np);
```

```
[media] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab de Meios\mmq\Resultados\lgeral\sem_outliers\media_geral_ímpar_sout.txt");
[L_linhas, L_colunas] = size(media);
```

```
[var] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab de Meios\mmq\Resultados\lgeral\sem_outliers\var_geral_ímpar_sout.txt");
[L_linhas, L_colunas] = size(var);
```

```
[dp] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab de Meios\mmq\Resultados\lgeral\sem_outliers\desvp_geral_ímpar_sout.txt");
```

```

[L_linhas, L_colunas] = size(dp);

[one] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab
de Meios\mmq\Resultados\l23\sem_outliers\one.txt");
[L_linhas, L_colunas] = size(one);

[media2] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab
de Meios\mmq\Resultados\lgeral\sem_outliers\media2_geral_ímpar_sout.txt");
[L_linhas, L_colunas] = size(media2);

y=np;
x=[media,media2,var,one];

//Calcula teta
teta = ([x']*x)^-1*x'*y;
disp("Teta calculado:");
disp (teta);

//Calculo dos resíduos

//resíduo simples
rs = y-(x*teta);
disp (rs, "residuo simples =");

//resíduo médio
rm= mean(rs);
disp (rm, "resíduo médio =");

//resíduo absoluto médio
ram = mean(abs(rs));
disp (ram, "resíduo absoluto médio =");

```

- Rotina para validação dos modelos e estimação de pessoas.

Modelo de Primeira Ordem

//Thiago Tortorelli de Faria

```

////////////////////////////////////
//////////Estimando o número de pessoas do modelo de 1 ordem//////////
////////////////////////////////////Link: Média Geral////////////////////////////////////

```

//Dados de entrada utilizados, índices ímpar de cada link e cada janela. São dados de validação.

clc();

```

[np] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab de
Meios\mmq\Resultados\l23\sem_outliers\npessoas.txt");
[L_linhas, L_colunas] = size(np);

```

```

[media] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab
de Meios\mmq\Resultados\lgeral\sem_outliers\media_geral_par_sout.txt");
[L_linhas, L_colunas] = size(media);

```

```

[var] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab de
Meios\mmq\Resultados\lgeral\sem_outliers\var_geral_par_sout.txt");
[L_linhas, L_colunas] = size(var);

```

```

[dp] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab de Meios\mmq\Resultados\lgeral\sem_outliers\desvp_geral_par_sout.txt");
[L_linhas, L_colunas] = size(dp);

[one] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab de Meios\mmq\Resultados\l23\sem_outliers\one.txt");
[L_linhas, L_colunas] = size(one);

[media2] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab de Meios\mmq\Resultados\lgeral\sem_outliers\media2_geral_par_sout.txt");
[L_linhas, L_colunas] = size(media2);

tetam=- 27.423506;
tetav=- 6.03965;
tetaconst=29.753614;

yestimado = media*tetam+var*tetav+tetaconst;
disp (yestimado,"Número estimado de pessoas do modelo de 1 ordem:");

//resíduo absoluto médio
ram=mean(abs(np-yestimado));
disp(ram,"residuo absoluto médio:");

ra=abs(np-yestimado);
disp(ra,"residuo absoluto:");

```

Modelo de Segunda Ordem

//Thiago Tortorelli de Faria

```

////////////////////
//////////Estimando o número de pessoas do modelo de 2 ordem//////////
////////////////////Link: Média Geral////////////////////

```

//Dados de entrada utilizados, índices ímpar de cada link e cada janela. São dados de validação.

```

//média = x1
//media2 = (x1)^2
//dp = x2
//var = (x2)^2
clc();

```

```

[np] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab de Meios\mmq\Resultados\l23\sem_outliers\npessoas.txt");
[L_linhas, L_colunas] = size(np);

```

```

[media] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab de Meios\mmq\Resultados\lgeral\sem_outliers\media_geral_imp_ar_sout.txt");
[L_linhas, L_colunas] = size(media);

```

```

[var] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab de Meios\mmq\Resultados\lgeral\sem_outliers\var_geral_imp_ar_sout.txt");
[L_linhas, L_colunas] = size(var);

```

```

[dp] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab de Meios\mmq\Resultados\lgeral\sem_outliers\desvp_geral_imp_ar_sout.txt");
[L_linhas, L_colunas] = size(dp);

```

```

[one] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab
de Meios\mmq\Resultados\l23\sem_outliers\one.txt");
[L_linhas, L_colunas] = size(one);

[media2] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes
Lab de Meios\mmq\Resultados\lgeral\sem_outliers\media2_geral_imp_ar_sout.txt");
[L_linhas, L_colunas] = size(media2);

tetam=266.47256;
tetam2=292.14402;
tetadp=- 24.270506;
tetav=18.104163;
tetaconst=- 258.552327;

yestimado = (media*tetam)+(media2*tetam2)+(dp*tetadp)+(var*tetav)+tetaconst;
disp (yestimado, "Número estimado de pessoas do modelo de 2 ordem:");

//resíduo absoluto médio
ram=mean(abs(np-yestimado));
disp(ram,"residuo absoluto médio:");

ra=abs(np-yestimado);
disp(ra,"residuo absoluto:");

```

Modelo de Terceira Ordem

//Thiago Tortorelli de Faria

```

////////////////////
//////////Estimando o número de pessoas do modelo de 3 ordem//////////
////////////////////Link: Média Geral////////////////////

```

//Dados de entrada utilizados, índices ímpar de cada link e cada janela. São dados de validação.

```

//média = x1
//media2 = (x1)^2
//dp = x2
//var = (x2)^2
clc();

```

```

[np] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab de
Meios\mmq\Resultados\l23\sem_outliers\npessoas.txt");
[L_linhas, L_colunas] = size(np);

```

```

[media] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab
de Meios\mmq\Resultados\lgeral\sem_outliers\media_geral_par_sout.txt");
[L_linhas, L_colunas] = size(media);

```

```

[var] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab de
Meios\mmq\Resultados\lgeral\sem_outliers\var_geral_par_sout.txt");
[L_linhas, L_colunas] = size(var);

```

```

[dp] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab de
Meios\mmq\Resultados\lgeral\sem_outliers\desvp_geral_par_sout.txt");
[L_linhas, L_colunas] = size(dp);

```

```

[one] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes Lab
de Meios\mmq\Resultados\l23\sem_outliers\one.txt");
[L_linhas, L_colunas] = size(one);

```

```
[media2] = fscanfMat("C:\Users\User\Google Drive\Dissertação\Nova Proposta 27-03-15\Testes  
Lab de Meios\mmq\Resultados\Igeral\sem_outliers\media2_geral_par_sout.txt");  
[L_linhas, L_colunas] = size(media2);
```

```
tetam=272.70042;  
tetam2=299.89125;  
tetav=- 4.5940959;  
tetaconst=- 268.8857;
```

```
yestimado = (media*tetam)+(media2*tetam2)+(var*tetav)+tetaconst;  
disp (yestimado, "Número estimado de pessoas do modelo de 3 ordem:");
```

```
//resíduo absoluto médio  
ram=mean(abs(np-yestimado));  
disp(ram, "resíduo absoluto médio:");
```

```
//resíduo absoluto  
ra=abs(np-yestimado);  
disp(ra, "resíduo absoluto:");
```

ANEXO D – Mapa do pacote Radiuino – 52 Bytes.

Cabeçalhos das Camadas	PhyHdr[#]				MacHdr[#]				NetHdr[#]				TranspHdr[#]					
Posição na variável	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3		
Função das posições do cabeçalho	RSSI Downlink	LQI Downlink	RSSI Uplink	LQI Uplink	Sleep	Tempo Sleep 1	Tempo Sleep 2	TDB	DST_ID Quem Recebe	DST_NID	SRC_ID Quem Manda	SRC_NID	COUNT	TDB	TDB	TDB		
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 das posições dos 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 das posições dos 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