

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

MATHEUS MAIA DE SOUZA

**Ferramenta para visualização de *logs* de
redes *WirelessHart***

Monografia apresentada como requisito parcial
para a obtenção do grau de Bacharel em
Engenharia da Computação

Orientador: Prof. Dr. João Cesar Netto

Porto Alegre
2016

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Rui Vicente Oppermann

Vice-Reitor: Prof. Jane Fraga Tutikian

Pró-Reitor de Graduação: Prof. Vladimir Pinheiro do Nascimento

Diretor do Instituto de Informática: Prof. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Engenharia de Computação: Prof. Raul Fernando Weber

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Agradeço a todos que contribuíram de forma direta ou indireta para a realização deste trabalho, em especial aos meus pais, Flávio e Mirian, que sempre incentivaram o caminho da educação.

Aos amigos e colegas que me acompanharam ao longo dos anos de formação acadêmica. Em especial aos colegas de laboratório Ivan Muller e Jean Winter que sempre estiveram dispostos a compartilhar seus conhecimentos.

Ao Professor João Cesar Netto pelos anos de orientação em diversos projetos. E aos demais professores que contribuíram para minha formação.

RESUMO

O método, já consolidado, que faz uso de cabos como meio de comunicação impõe algumas restrições como por exemplo, alto custo para reestruturação e cuidado constante com a manutenção e troca de cabos. O protocolo *WirelessHART* contorna estas restrições oferecendo uma solução sem fio. Após anos de pesquisa o protocolo *WirelessHART* se mostra uma alternativa robusta e confiável para ambientes tão desafiadores como o chão de fábrica. Porém, ainda existe a necessidade da criação de ferramentas de suporte em certas áreas, como por exemplo, na fase de testes, planejamento e instalação da rede. Neste trabalho será apresentado um resumo da organização do protocolo e suas funcionalidades e uma proposta de ferramenta para visualização de dados vindos de capturas realizadas em redes *WirelessHART*. A ferramenta lê os arquivos de captura e obtém as conexões entre os dispositivos da rede para, em uma interface gráfica, exibir a topologia da rede. Navegando pela topologia, o usuário pode obter as informações pertinentes aos dispositivos e estatísticas das suas conexões. Essa visão macroscópica permite uma análise mais eficiente dos dados, evitando buscas manuais linha-a-linha em arquivos extensos .

Palavras-chave: WirelessHart, Visualização, Wi-Analys.

ABSTRACT

The consolidate method that uses cables as communication medium imposes some restrictions, for example, high restructuring cost, a constant maintenance and cable changes. WirelessHART get around this restrictions offering a wireless solution. After years of research the protocol show itself a robust and reliable alternative for a challenging environment as factory floor. However, there is a need for new support tools in certain areas, for instance, test, planning and setup phase. This work will be showing a WirelessHART protocol overview, its features and organization. Also will be proposed an visualization tool for data logs coming from WirelessHART networks sniffing. This tool reads the log file and obtain the connections between the network devices to display the network topology in a graphical interface. By navigating through the topology the user can obtain the pertinent information to the devices and their connections Statistics . This macroscopic view allows for a more efficient analysis of the data, avoiding line-by-line manually searches in big files.

Keywords: *WirelessHart*, Visualization, Wi-Analys.

LISTA DE ABREVIATURAS E SIGLAS

OSI	<i>Open Systems Interconnection</i>
ISM	<i>Industrial Scientific and Medical</i>
EIRP	<i>Equivalent isotropically radiated power</i>
CCA	<i>Clear Channel Assessment</i>
LLC	<i>Logical Link Control</i>
MAC	<i>Media Access Control</i>
DLPDU	<i>Data Link Protocol Data Unit</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
AES	<i>Advanced Encryption Standard</i>
WH	<i>WirelessHart</i>
WPF	<i>Windows Presentation Foundation</i>
XAML	<i>Extensible Application Markup Language</i>
RSL	<i>Received Signal Level</i>
FPGA	<i>Field Programmable Gate Array</i>

LISTA DE FIGURAS

Figura 2.1 Modelo de camadas OSI. Adpatada de (CHEN; NIXON; MOK, 2010).	11
Figura 2.2 <i>Data-Link Protocol Data Unit</i> (DLPDU). Adaptada de (SPEC-075, 2008).	14
Figura 2.3 Estrutura <i>superframe</i> . Adaptada de (SPEC-075, 2008).	15
Figura 2.4 Sequência de transmissão em um <i>time slot</i> . Adaptada de (SPEC-075, 2008).	15
Figura 3.1 Fluxo de trabalho de teste e análise de rede WH.....	18
Figura 5.1 Captura de tela- <i>Wi-Analys</i>	23
Figura 7.1 Diagrama de casos do Visualizador de <i>logs WirelessHart</i>	26
Figura 7.2 Diagrama de classes do Visualizador de <i>logs WirelessHart</i>	27
Figura 7.3 Diagrama de sequência- Buscar <i>logs WirelessHart</i>	30
Figura 7.4 Diagrama de sequência-Obter dados de Arquivo.	31
Figura 7.5 Diagrama de sequência-Inserir Plano de Fundo.	31
Figura 7.6 Diagrama de sequência-Montar estruturas,comando 962.....	32
Figura 7.7 Diagrama de sequência-Montar estruturas, comando 968.....	33
Figura 7.8 Diagrama de sequência-Montar estruturas, comando 967.....	33
Figura 7.9 Diagrama de sequência-Montar estruturas, comando 780.....	34
Figura 7.10 Diagrama de sequência-Montar estruturas, comando 787.....	35
Figura 7.11 Diagrama de sequência-Avançar <i>Snap</i>	35
Figura 7.12 Diagrama de sequência-Retornar <i>Snap</i>	36
Figura 8.1 Controles da ferramenta de visualização de <i>logs Wi-Analys</i>	37
Figura 8.2 Menu da ferramenta de visualização de <i>logs Wi-Analys</i>	38
Figura 8.3 Painel de informações detalhadas da ferramenta de visualização de <i>logs</i> <i>Wi-Analys</i>	39
Figura 8.4 Exemplo de topologia de rede	41
Figura 8.5 Fragmento de Log <i>Wi-Analys</i> 1	44
Figura 8.6 Visualização distribuição de links-Cenário 1	45
Figura 8.7 Fragmento de Log <i>Wi-Analys</i> 2	46
Figura 8.8 Visualização dados de qualidade <i>EAWERE-01</i> - Cenário 2.....	47
Figura 8.9 Visualização dados de qualidade <i>EAWERE-02</i> - Cenário 2.....	47

LISTA DE TABELAS

Tabela 2.1	Valores esperados de distância vs potência.	17
Tabela 8.1	Classificação de cores de acordo com a proporção de falhas	42

SUMÁRIO

1 INTRODUÇÃO	10
2 WIRELESSHART: UMA VISÃO GERAL	11
2.1 Camada de Aplicação	12
2.2 Camada de Rede e Transporte	12
2.3 Camada de Enlace	13
2.3.1 Logical Link Control (LLC)	13
2.3.2 <i>Medium Access Control (MAC)</i>	14
2.4 Camada Física	16
3 OBJETIVOS	18
4 TRABALHOS RELACIONADOS	20
5 FERRAMENTA DE CAPTURA DE REDE – SNIFFER WI-ANALYS	22
6 DEFINIÇÕES DE PROJETO DO VISUALIZADOR DE LOGS WIRELESSHART	24
6.1 Linguagem de programação	24
6.2 Bibliotecas externas	24
7 METODOLOGIA DE PROJETO-VISUALIZADOR DE LOGS WIRELESSHART	25
7.1 Definição dos casos de uso	25
7.2 Diagrama de Classes	27
7.3 Diagramas de sequência	29
7.3.1 <i>Buscar logs</i>	29
7.3.2 <i>Obter dados de Arquivo</i>	29
7.3.3 <i>Inserir Plano de Fundo</i>	30
7.3.4 <i>Montar estruturas</i>	31
7.3.4.1 <i>Write Device Nickname Address (962)</i>	32
7.3.4.2 <i>Write Link (968) e Delete Link (967)</i>	33
7.3.4.3 <i>Report Neighbor Heath List (780)</i>	34
7.3.4.4 <i>Report Neighbor Signal Levels (787)</i>	34
7.3.5 <i>Avançar Snap/ Retornar Snap</i>	34
8 RESULTADOS OBTIDOS	37
8.1 Apresentação e aspectos gerais.	37
8.1.1 <i>Controles</i>	37
8.1.2 <i>Informações de detalhadas</i>	38
8.1.3 <i>Topologia</i>	40
8.2 Utilização	42
8.2.1 <i>Cenário 1</i>	42
8.2.2 <i>Cenário 2</i>	43
9 CONCLUSÃO	48
REFERÊNCIAS	49
ANEXO A — TRABALHO DE GRADUAÇÃO - I	50
ANEXO B — VISUALIZADOR DE LOGS WI-ANALYS	62
ANEXO C — GUIA DE UTILIZAÇÃO DO SOFTWARE	64

1 INTRODUÇÃO

O processo de automatização industrial é essencial para o aumento da produtividade e redução de custos. Sensores e atuadores são conectados a redes para monitorar e gerenciar os dispositivos. A solução clássica são redes que utilizam cabos para transmitir as informações, estratégia que torna mais cara e menos flexível a implementação de redes de comunicação para ambientes industriais. Com o surgimento do protocolo WirelessHART (WH), primeira tecnologia de comunicação sem fio certificada para uso industrial, muitas vantagens em relação aos tradicionais sistemas cabeados surgiram: redução de custos com infraestrutura, possibilidade de instalação em locais de difícil acesso, redução do tempo de instalação, entre outros. Entretanto, tal tecnologia ainda carece de ferramentas que facilitem o desenvolvimento e teste de novos dispositivos. Com a intenção de preencher esta lacuna será apresentado neste trabalho o projeto de um programa para auxiliar a análise de dados obtidos em testes de redes WH.

O programa desenvolvido é um visualizador de dados. Os visualizadores de dados cumprem um papel importante na fase de análise, pois ampliam as percepções dos analistas gerando um ganho de eficiência e qualidade nas análises. Antes da implementação da ferramenta algumas tarefas chaves foram modeladas, para guiar e documentar a etapa de codificação. Por fim, tendo a ferramenta operacional, são relatados os resultados obtidos na forma de exemplos de uso.

2 WIRELESSHART: UMA VISÃO GERAL

O protocolo de comunicação *HART* está consolidado no setor de automação industrial com aproximadamente trinta milhões de dispositivos instalados (GROUP, 2014). Apesar disso, e a fim de estender as capacidades desse protocolo, a *HART Communication Foundation* apresentou na revisão de número sete alterações objetivando comunicação sem fio: o *WirelessHART*. Assim como nas revisões anteriores, a revisão sete tem como objetivo o controle de atuadores, aquisição de variáveis de processo e gerenciamento de ativos. As alterações trazidas pela versão sete trouxeram maior flexibilidade ao protocolo sem perder os recursos instalados. O uso de adaptadores permite que redes já consolidadas possam ser integradas a novas redes sem fio.

A arquitetura do protocolo *WirelessHART* é descrita através do modelo de camadas OSI (*Open Systems Interconnection*) (CHEN; NIXON; MOK, 2010). A figura 1 mostra a organização das camadas do protocolo com uma breve descrição da funcionalidade de cada camada.

Figura 2.1: Modelo de camadas OSI. Adpatada de (CHEN; NIXON; MOK, 2010).

Camada OSI	Função	WirelessHART
Aplicação	Provê ao usuário aplicações com acesso a rede.	Otimização de energia; Caminhos redundantes; Rede em malha auto-organizável.
Apresentação	Converte os dados da aplicação entre os formatos de rede e formatos locais da máquina.	
Sessão	Conexão e gerenciamento dos serviços para aplicações	
Transporte	Provê independência de rede. Transferência de mensagens transparente.	Transferência de conjunto grande de dados Auto-segmentada; Fluxo de transporte confiável; Tamanhos de segmentos negociáveis.
Rede	Roteamento fim a fim dos pacotes. Resolver os endereços de rede.	Otimização de energia; Caminhos redundantes; Rede em malha auto-organizável.
Enlace	Estabelece a estrutura dos pacotes. Enquadramento. Detecção de erros.	Segurança; Sincronização; TDMA / CSMA. Salto de Canais; Two-way Handshake.
Física	Conexão elétrica e mecânica. Transmissão bruta fluxo de bits.	Rádios baseados em IEEE 802.15.4 (2.4GHz); Potência máxima de transmissão de 10dBm

A seguir cada camada e outras características do protocolo serão discutidas.

2.1 Camada de Aplicação

É na camada de aplicação que os comandos de processo são implementados, definindo portanto o comportamento dos comandos e os tipos de dados que devem ser carregados nas mensagens. Cada comando especifica de forma única e não ambígua o pacote de dados e seu tamanho.

Os comando são divididos nas classes:

- ***Universal Commands***: São comandos que devem ser suportados por todos dispositivos *HART*. (SPEC-127, 2008);
- ***Common Practice Commands***: Comandos implementados por um grande número de dispositivos e devem ser suportados sempre que possível. (SPEC-151, 2008);
- ***Non-public Commands***: Comandos especiais que têm seu uso limitado para ambiente de desenvolvimento dos dispositivos;
- ***Wireless Commands***: São a base da comunicação das redes *WirelessHART* sendo mandatório para produtos *WirelessHART*. (SPEC-155, 2008);
- ***Device Family Commands***: Grupo de comandos usados para configuração de dispositivos de campo, sem a necessidade de implementações individualizadas;
- ***Device-Specific Commands***: Comandos definidos pelos fabricantes segundo as necessidades e propósitos de seus dispositivos.

2.2 Camada de Rede e Transporte

São funcionalidades (SPEC-085, 2009) das camadas, Rede e Transporte: .

- Endereçamento e entrega de dados;
- Comunicação confiável;
- Segmentação;
- Controle de erros;
- Estabelecimento da conexão entre dois nós da rede.

São dois os modelos de roteamento das mensagens previstos na norma: Grafo ou *Source*. O primeiro utiliza uma estrutura de grafos para representar um conjunto de dispositivos e links entre dispositivos, esses grafos descrevem rotas redundantes entre o emissor e o receptor. As condições da rede quando um pacote é enviado definem por qual

caminho o mesmo deve seguir. O segundo é estático e especificado no próprio pacote. Esse caminho é único entre a origem e destino, então se em algum ponto deste caminho houver falha o pacote é perdido. Nos casos de falha o dispositivo que detectou a falha notifica o gerenciador de rede para devidas correções.

A camada de rede gerencia as comunicações fim-a-fim utilizando sessões criptografadas. Para isto, os dispositivos devem manter informações de segurança tais como chaves de encriptação. O padrão utilizado é o AES 128 bit, com dois tipos de chaves: *Join Key* e *session key*. A *Join Key* deve ser configurada previamente para que o dispositivo esteja habilitado a entrar na rede. A *session key* é configurada após a entrada do dispositivo e previne que ataques maliciosos possam injetar dados falsos nos fluxos válidos.

2.3 Camada de Enlace

A camada de enlace ou *Data link layer* deve garantir a transferência dos dados entre os dispositivos da rede, para isso dispõe de recursos de detecção e correção de erros e de mecanismos de controle de acesso ao meio. O recurso de controle de acesso ao meio visa garantir que a qualidade da comunicação não seja reduzida por colisões entre quadros WH. As responsabilidades são divididas em duas subcamadas: *Logical Link Control* (LLC) e *Media Access Control* (MAC). (SPEC-075, 2008).

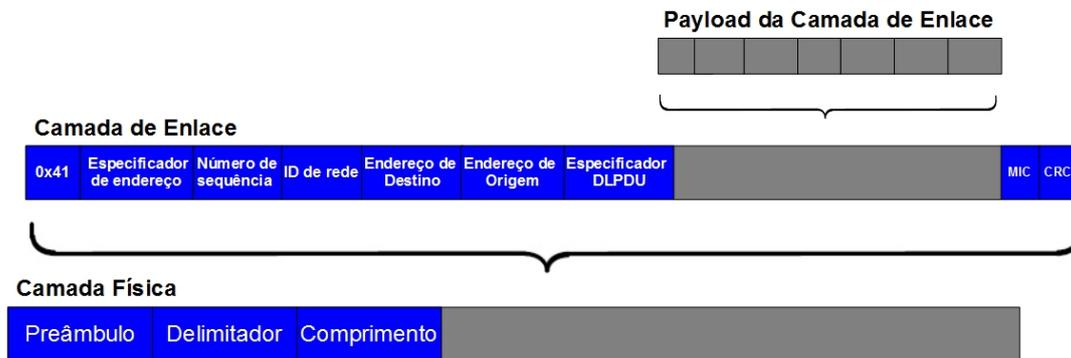
2.3.1 Logical Link Control (LLC)

A LLC é responsável pelo controle de erros, controle de fluxo e montagem do quadro WH.

O cabeçalho do quadro da camada de enlace é composto pelos campos:

- **Byte de controle:** (0x41) indica o início do quadro;
- **Especificador de endereço:** define qual tipo de endereço está presente no quadro. Os endereços podem ser de dois tipos: um *nickname* de dois *bytes* ou um IEEE EUI-64 de oito *bytes*;
- **Número de sequência:** indica a sequência dos quadros;
- **ID de rede:** define a qual rede o quadro pertence.
- **Endereços Origem/Destino:** contém os endereços dos dispositivos envolvidos na

Figura 2.2: *Data-Link Protocol Data Unit (DLPDU)*. Adaptada de (SPEC-075, 2008).



comunicação;

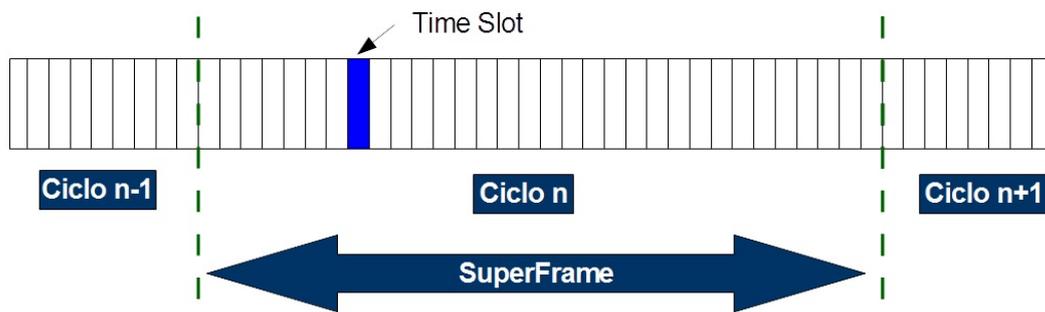
- **Campo especificador DLPDU:** tem função no controle de fluxo do protocolo. utilizando um *byte* codifica três características do quadro WH, prioridade, *network key* e tipo de pacote.
 - **Prioridade:** dividida em quatro níveis, *Command*, *Process Data*, *Normal*, *Alarm*. Sendo o primeiro nível citado o mais prioritário seguido pelos demais na ordem em que foram dispostos;
 - **Network Key:** Define tipo de chave criptográfica utilizada na transmissão (*network key* ou *public key*);
 - **Tipos de mensagens:** classificados como *ACK*, *Advertise*, *Keep Alive*, *Disconnect*, *Data*.

No *byte* especificador DLPDU os dois primeiros bits mais significativos são reservados para futuras aplicações.

2.3.2 Medium Access Control (MAC)

Para evitar colisões na troca de mensagens e garantir uma comunicação determinística, o protocolo *WirelessHART* emprega a técnica TDMA. A principal estrutura utilizada nesta técnica é o *superframe*. A figura 2.3 mostra a estrutura do *superframe*.

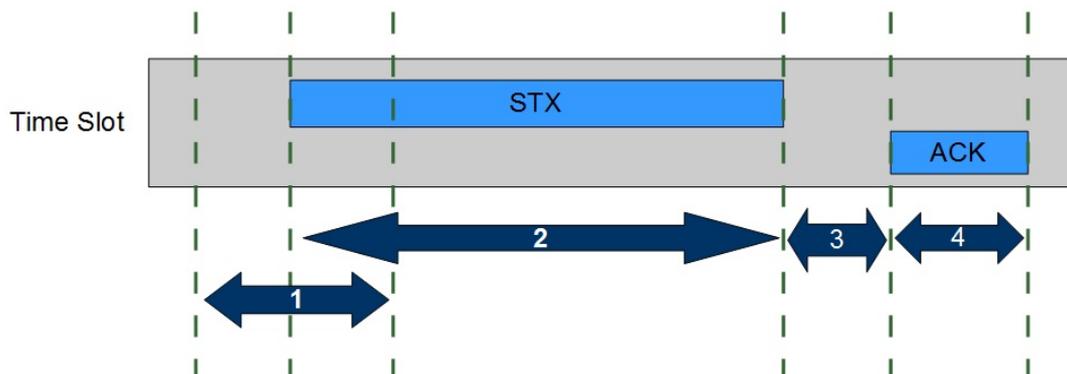
Figura 2.3: Estrutura *superframe*. Adaptada de (SPEC-075, 2008).



O *time slot* é base para o *superframe* e divide sua estrutura em intervalos de tempos bem definidos. Cada *time slot* é associado a uma comunicação entre dispositivos ou a uma comunicação do tipo *broadcast*. Uma transmissão de um dispositivo X para um dispositivo Y, por exemplo, é associada a um ou mais *time slot* e só poderá ocorrer nos tempos determinados. Quando transcorrido o tempo do último *time slot* inicia-se um novo ciclo do *superframe*.

A figura 2.4 mostra a sequência da transmissão, em condições normais, entre dispositivos:

Figura 2.4: Sequência de transmissão em um *time slot*. Adaptada de (SPEC-075, 2008).



No intervalo 1 receptor espera o início da transmissão, seguindo em 2 ocorre o envio dos dados por parte do transmissor ao receptor. Em 3 o transmissor espera pelo envio da mensagem de confirmação *ack*. Em 4 o receptor envia a mensagem de *ack* e a comunicação é encerrada. Mensagens do tipo *broadcast* não recebem *ack*.

A subcamada MAC tem como objetivos:

- Manter a sincronização dos tempos de comunicação;
- Identificar os *slots* de tempo que devem ser utilizados;

- Propagar mensagens vindas de vizinhos;
- Repassar para a camada física os quadros vindos da camada de rede.

2.4 Camada Física

A camada física tem como objetivo: fornecer serviços para a camada de dados e controlar a interação entre dispositivo e meio de transmissão. Para isto utiliza parâmetros bem definidos nos documentos da norma. Para o tráfego das informações é utilizada a faixa de frequência ISM 2,4 GHz. (SPEC-065, 2007)

A lista a seguir descreve um conjunto de características da camada física.

- **Tempo entre *time slots*:** em uma rede *WirelessHART* a menor janela de tempo para uma comunicação é 10 ms. Quando em modo *unicast* a mensagem de requisição e de confirmação devem ser enviadas dentro da janela de tempo. Já em *broadcast* apenas uma mensagem é enviada dentro da janela de tempo uma vez que as mensagens de *broadcast* não recebem confirmação;
- **Tempos entre mensagens no *time slot*:** o menor tempo entre duas mensagens é 1 ms, tempo entre o fim da transmissão e o início do envio de confirmação por parte do receptor;
- **Quadro IEEE 802.15.4:** todas mensagens trocadas em uma rede WH utilizam o mesmo tipo de dados definidos na norma IEEE 802.15.4;
- **Canais:** são utilizados 15 canais (11-25) da ISM, o canal 26 não é utilizado por ter seu uso não autorizado em muitas localidades;
- **Interferências:** para evitar interferências é utilizada a técnica de salto de canais a cada transmissão;
- **Colisões:** as colisões entre transmissões simultâneas são mitigadas pela técnica de CCA que analisa o meio antes de transmitir a mensagem: caso o meio já esteja ocupado o dispositivo deve aguardar nova oportunidade para transmitir;
- **Sensibilidade do dispositivo:** normalizada, sendo exigido para todo o sistema (rádio, circuito, conectores e etc) o valor de -85 dBm ou melhor;
- **Potência de transmissão:** Todos dispositivos devem ter um EIRP (*Equivalent isotropically radiated power*) de até +10dBm (10mW) com tolerância de +/-3dB. A faixa programável para a potência deve ser -10dBm a +10dBm;

- **Tabela de valores esperados para o alcance (2.1):** Os valores são estimados assumindo ganho unitário em uma antena omnidirecional, intensidade de recepção de -82dBm e ambiente sem interferência.

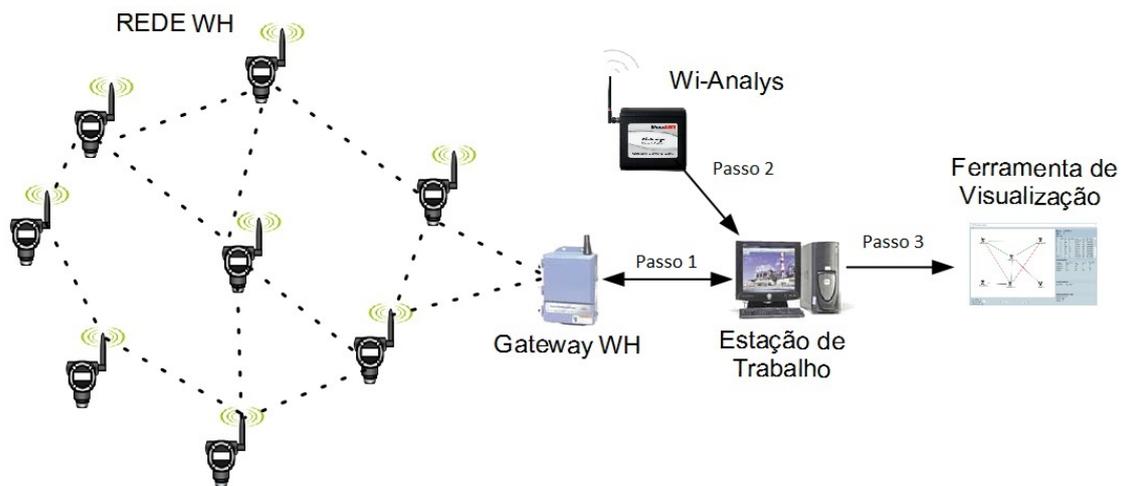
Tabela 2.1: Valores esperados de distância vs potência.

Ambiente	Potência	Distância (em metros)
Externo (com visada)	0 dBm	100
Externo (com visada)	10 dBm	200
Interno	0 dBm	35
Interno	10 dBm	75

3 OBJETIVOS

Antes de definir o objetivo deste trabalho deve-se entender o problema a ser resolvido. Para apresentar o problema, inicia-se com a descrição do o fluxo de trabalho de teste e análise de uma rede WH (figura3.1).

Figura 3.1: Fluxo de trabalho de teste e análise de rede WH.



O fluxo de trabalho apresentado na figura (figura3.1) é descrito a seguir. Passo 1: comandos de controle são enviados. As repostas desses comandos e os dados do monitoramento das variáveis de processo são recebidos. Passo 2: ao fim do teste, os dados capturados pela ferramenta *Wi-Analys* são salvos na estação de trabalho. Passo 3: o arquivo *log* da ferramenta *Wi-Analys* é analisado, para se verificar o comportamento da rede quando submetida às ocorrências do teste. Atualmente, o passo 3 pode ser realizado tanto com o software mostrado em 5 como com o uso de manipuladores de tabelas. Porém, ambos apresentam problemas que dificultam a eficiência da análise. Devido ao longo tempo de execução dos testes, os arquivos gerados são grandes ao ponto de não serem bem suportados pelas ferramentas existentes. Um segundo fato que gera dificuldade vem da percepção humana, que é mais eficiente no entendimento de imagens ou modelos do que no entendimento de textos e códigos. Um exemplo que pode ser destacado é para a geração da topologia de uma rede WH. Utilizando os manipuladores de tabelas o analista deve percorrer milhares de linhas buscado a geração e exclusão de links entres os dispositivos e então montar a topologia de forma manual.

O objetivo da ferramenta apresentada neste trabalho é aumentar a eficiência das

análises oferecendo uma visão macro dos dados obtidos nos *logs* do *Wi-Analys*. Assim, o analista pode usar o ganho de tempo para focar nos problemas ou detalhes importantes detectados com o uso do visualizador.

4 TRABALHOS RELACIONADOS

Esta seção apresenta trabalhos relacionados a ferramentas de avaliação, em diferentes aspectos, de redes WirelessHART.

Em (LORENÇATO, 2013) o autor descreve duas ferramentas para coleta de dados de redes WH. Coletor de Dados Monocanal: essa ferramenta é composta por dois elementos, o coletor de dados e o gerenciador da unidade de coleta de dados. A unidade de coleta de dados utiliza como base um dispositivo WH, mantendo funções como a sintonização dos canais. Já o gerenciador por sua vez é responsável pela coleta e pelo processamento das informações dos *links* de comunicação. Conhecendo o *time slot*, número de *superframe* e canal em que acontecerá a comunicação o coletor é capaz de capturar as mensagens.

A segunda proposta utiliza um hardware capaz de capturar 16 canais simultaneamente. Esta abordagem dispensa a necessidade da unidade de gerenciamento e de cuidados com os tempos de sincronização. Porém, após a coleta os dados devem ser processados para que sejam colocados em ordem cronológica. Os dados então são apresentados em uma interface gráfica para o usuário.

Em (MACHADO, 2014) é descrita uma ferramenta que realiza o sensoriamento da rede e apresenta estatísticas a respeito da mesma. São utilizadas duas peças de hardware, uma para captura dos dados e outra para armazenar informações em um cartão de memória. Além das estatísticas, a ferramenta possui função de detecção de coexistência da rede WH com outras fontes que atuam na mesma faixa de frequência.

Em (KUNZEL, 2012), o autor propõe um ambiente para avaliação de estratégias de roteamento. Os algoritmos de roteamento, de uma forma geral, buscam uma rota que minimize alguma função de custo, como por exemplo, o número de saltos ou consumo de energia. Os dados capturados são enviados posteriormente para o software que analisa os registros e apresenta os links entre dispositivos e valores de RSL utilizando grafos.

Em (FERRARI et al., 2010) o autor apresenta uma ferramenta para captura e análise de redes WH. O conjunto conta com rádios receptores, chamados de sondas, conectados à um FPGA, que realiza a sincronização dos quadros WH e o encapsulamento das informações em pacotes UDP/IP para o envio para estação de trabalho via *Ethernet*. Toda comunicação WH é decifrada em modo *off-line* na estação de trabalho. Para realizar a análise os autores desenvolveram um *dissector* para a ferramenta *WireShark*. Essa funcionalidade permite que o *WireShark* compreenda os formatos dos pacotes WH e extraia

estatísticas da rede WH analisada.

Em (LEAL, 2016) o autor descreve o sistema de testes de conformidade, fornecido pela *HART Foundation*, e apresenta uma ferramenta para facilitar e agilizar o trabalho dos analistas que atuam na fase de testes de conformidade. A partir dos *logs* vindos das ferramentas *Htest Post Proccessing Suite* e *Wi-Analys*, que possuem uma grande quantidade de informações, a ferramenta cria um terceiro arquivo de *log* com informações mais simplificadas sobre os resultados obtidos nos testes. Para gerar esse terceiro arquivo, a ferramenta percorre os *logs* de entrada capturando campos de interesse e realizando a correspondência de pacotes, entre os *logs Htest Post Proccessing Suite* e *Wi-Analys*.

5 FERRAMENTA DE CAPTURA DE REDE – *SNIFFER WI-ANALYS*

O *Wi-Analys* é uma ferramenta, composta por hardware e software, utilizada para capturar os pacotes que trafegam pela rede WH. Com as informações contidas nos pacotes capturados é possível realizar testes de conformidade e outros diagnósticos para certificar dispositivos WH.

O dispositivo pode receber até 16 canais simultaneamente e captura todas mensagens do padrão IEEE 802.15.4. O receptor *Wi-Analys* possui uma única antena física, utilizada para todos canais.

Para que o dispositivo possa decifrar as mensagens é necessário que o mesmo conheça as chaves de criptografia. Isto faz com que seja necessário o início do sensoramento antes que os dispositivos se conectem à rede. Já que as chaves de sessão são trocadas quando o dispositivo entra na rede.

Após garantir que o quadro é do tipo WH e realizar a decriptação das informações coletadas são apresentadas separadas em colunas, cada uma delas descreve um campo do quadro WH. A figura mostra a interface do componente de software da ferramenta *Wi-Analys*, utilizado para a apresentação do resultado da captura.

O software pode tanto apresentar os quadros que estão sendo capturados no momento do teste quanto apresentar os dados de *logs* obtidos em testes anteriores. É possível aplicar filtros para ocultar certo tipo de mensagem, um bom uso desta funcionalidade é ocultar as mensagens de *advertisement* que os dispositivos enviam via *broadcast* para que novos dispositivos ingressem na rede, como mostrado em 5.1. Ocultar esse tipo de mensagem facilita a visualização de mensagens mais relevantes.

A aplicação possui também a opção de exportar os dados em arquivos que podem ser lidos e editados por ferramentas de processamento de planilhas.

Apesar da filtragem poder ser realizada, resta um grande e complexo volume de dados que dificultam a análise. Utilizar a ferramenta *Wi-Analys* para uma análise, mesmo que simples, é necessário que o usuário tenha bom conhecimento da norma para realizar um diagnóstico adequado da rede.

Figura 5.1: Captura de tela-Wi-Analys.

Packet Number	Elapsed Time	RSL	Channel	PDU	Net ID	To	From	Graph ID	Dest	Src	Payload
256855	10,292,963.438	-45	17	Data	1236	0001	AIMATWILL	0000	F980	001B1E 0102 030405	40 00 00 [Response] 0, 23, 0, FE 20, 33, 0, 41 00 00 1 787, 7, 0, 00
256856	10,292,968.538	-43	17	ACK	1236	AIMATWILL	0001				
256893	10,293,963.432	-50	13	Data	1236	AIMATWILL	0001	0000	001B1E 0102 030405	F980	8F 00 00 961, 16, 91 962, 2, 00 963, 29, 00
256894	10,293,968.086	-39	13	ACK	1236	0001	AIMATWILL				
256923	10,294,963.452	-48	17	Data	1236	0001	AIMATWILL	0000	F980	000B	CF 00 00 [Response] 961, 17, 0, 31 962, 3, 0, 00 963, 30, 0, 00
256924	10,294,967.656	-43	17	ACK	1236	AIMATWILL	0001				
256965	10,295,963.444	-61	13	Data	1236	AIMATWILL	0001	0000	000B	F980	90 00 00 965, 5, 00 967, 8, 00 969, 5, 00 974, 5, 03 969, 4, 00 971, 3, 00
256966	10,295,967.455	-39	13	ACK	1236	0001	AIMATWILL				
257039	10,298,163.460	-51	17	Data	1236	0001	AIMATWILL	0080	F980	000B	D0 00 00 [Response] 967, 11, 0, 00 967, 11, 0, 00 974, 8, 0, 03 969, 6, 0, 00 971, 4, 0, 00
257040	10,298,167.665	-45	17	ACK	1236	AIMATWILL	0001				
257066	10,299,163.459	-45	13	Data	1236	AIMATWILL	0001	0000	000B	F980	91 00 00 963, 29, 00 974, 5, 02 969, 4, 00
257067	10,299,167.534	-39	13	ACK	1236	0001	AIMATWILL				
257104	10,300,163.469	-49	17	Data	1236	0001	AIMATWILL	0080	F980	000B	D1 00 00 [Response] 963, 30, 0, 00 974, 8, 0, 02 969, 6, 0, 00
257105	10,300,167.497	-43	17	ACK	1236	AIMATWILL	0001				
257168	10,302,163.480	-50	17	Data	1236	0001	AIMATWILL	0000	F980	000B	90 00 00 799, 9, 00
257169	10,302,166.183	-44	17	ACK	1236	AIMATWILL	0001				
257200	10,303,163.478	-46	13	Data	1236	AIMATWILL	0001	0000	000B	F980	D0 00 00 [Response] 799, 11, 0, 00
257201	10,303,166.500	-38	13	ACK	1236	0001	AIMATWILL				
257238	10,304,163.490	-46	17	Data	1236	0001	AIMATWILL	0080	F980	000B	43 00 00 [Response] 1, 5, F6, 01
257239	10,304,166.640	-43	17	Data	1236	AIMATWILL	0001				

6 DEFINIÇÕES DE PROJETO DO VISUALIZADOR DE LOGS WIRELESSHART

Antes da construção da ferramenta, algumas questões foram decididas com o objetivo de delimitar o escopo da aplicação e guiar a programação. Foram definidas as seguintes condições de contorno para o projeto.

6.1 Linguagem de programação

Por se tratar de uma aplicação cujo objetivo é apresentar os dados de uma forma mais clara faz-se necessária a utilização de elementos gráficos para representar estados e informações da rede e seus dispositivos. A linguagem C# possui *frameworks* que facilitam a construção de interfaces de usuário. O *framework* escolhido para o desenvolvimento foi o *Microsoft .NET*, mais especificamente o componente WPF - *Windows Presentation Foundation*.

O WPF separa a aparência da interface do seu comportamento: enquanto as características gráficas são definidas em um arquivo XAML-*Extensible Application Markup Language* os comportamentos são especificados em outro arquivo utilizando C#. A ligação entre interface e comandos se dá através de eventos gerados automaticamente pela interface.

6.2 Bibliotecas externas

Para auxiliar a construção do layout gráfico foi utilizada a biblioteca *Graph#*. A *Graph#* possui estruturas em forma de grafo que guardam as informações dos dispositivos e suas interligações, também responsável pela representação gráfica da estrutura. A biblioteca possui controles que são facilmente integrados com o *framework* WPF.

7 METODOLOGIA DE PROJETO-VISUALIZADOR DE LOGS WIRELESSHART

A metodologia empregada no desenvolvimento do projeto foi a modelagem utilizando a linguagem UML (RUMBAUGH; JACOBSON; BOOCH, 2004). Os modelos descrevem as principais funcionalidades do software e guiam o desenvolvimento do código que implementa tais funcionalidades. A modelagem, apesar de consumir um tempo importante do desenvolvimento da ferramenta, permite que na fase de codificação o programador se preocupe somente com aspectos de implementação, como se fora um tradutor. Então, o tempo dispendido no início do projeto converte-se em um investimento, já que o programador não usará seu tempo pensando na melhor forma de estruturar a software.

As etapas de modelagem a seguir serão apresentados neste capítulo:

- Definição dos casos de uso;
- Diagrama de classes;
- Diagramas de sequência.

7.1 Definição dos casos de uso

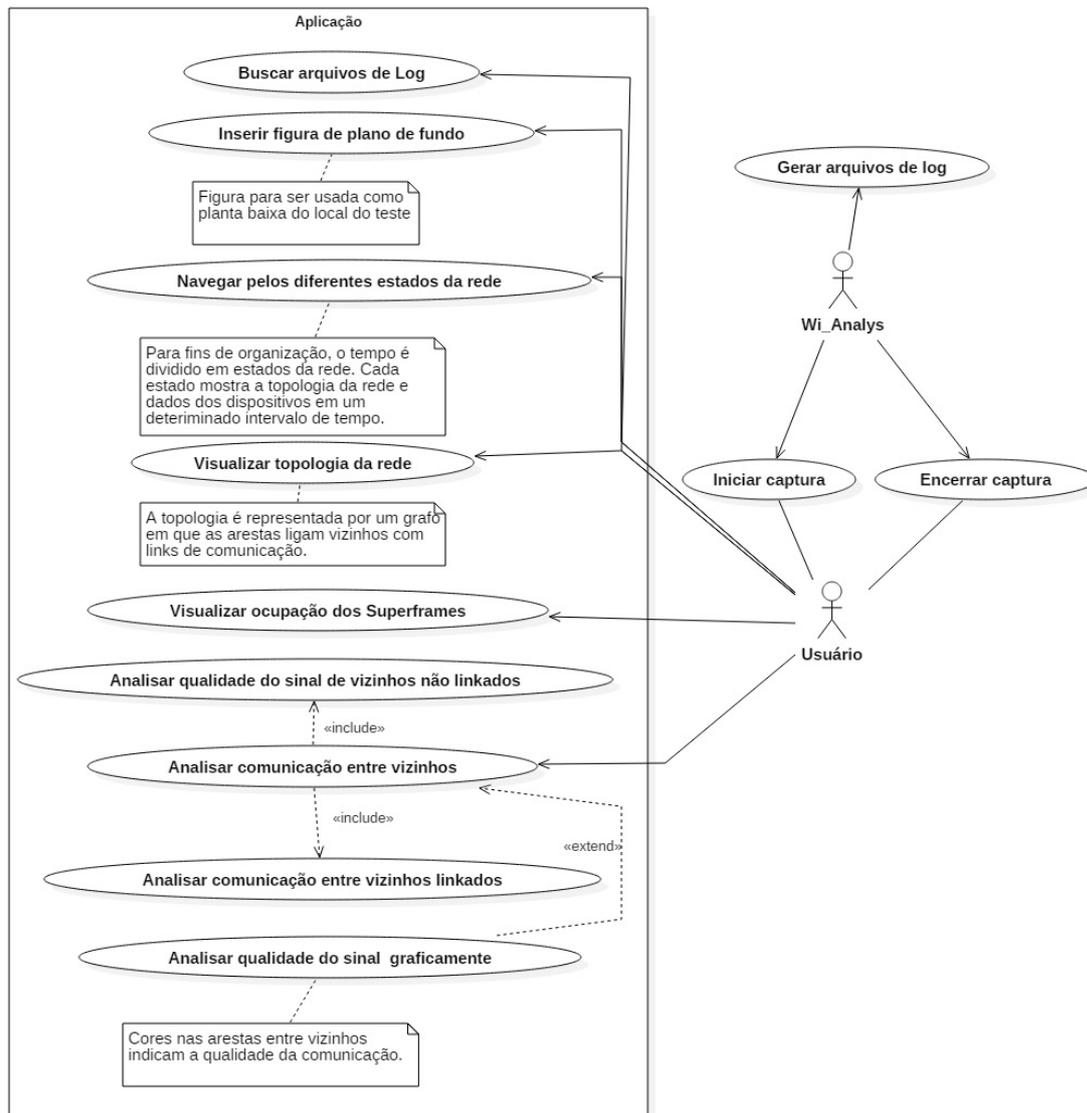
Esta fase do projeto documenta o ponto de vista do usuário, demonstrando as principais funcionalidades do software e interações dos usuários com o sistema.

Para o projeto foi elaborado o diagrama de casos de uso mostrado na figura 7.1 :

O diagrama mostra os atores Usuário e *Wi_Analys* e as associações entre funções e atores. A seguir uma breve descrição de cada funcionalidade do sistema:

- **Iniciar Captura:** esta deve ser a primeira atividade do usuário, devido às limitações da captura discutidas no capítulo: 5;
- **Gerar Arquivos de Log:** funcionalidade exclusiva do sistema *Wi_Analys*, gera o arquivo com as informações necessárias para a análise;
- **Buscar Log:** Primeira interação entre o usuário e o software de visualização. Antes de escolher o arquivo de entrada o usuário deve definir a granularidade de tempo em que o arquivo de entrada será dividido;
- **Inserir Figura de Plano de Fundo:** essa funcionalidade permite que o usuário insira uma imagem de fundo que contenha informações sobre o local onde foi realizado o experimento de captura dos dados. Informações como por exemplo: objetos

Figura 7.1: Diagrama de casos do Visualizador de logs WirelessHart.



que bloqueiam ou causem interferência no sinal de um dispositivo podem ajudar a entender o comportamento das transmissões;

- **Navegar pelos diferentes estados da rede:** ao ser dividido em intervalos de tempo, de mesma duração, o arquivo *log Wi-Analys* acaba gerando uma espécie de fotografia do estado da rede. O usuário pode então navegar por estes estados observando a evolução da rede ao longo do tempo;
- **Visualizar ocupação dos *Superframes*:** A ocupação dos *superframes* é exibida junto às informações dos links. Em cada link é descrito a qual *superframe* e *time slot* o mesmo pertence;
- **Visualizar topologia da rede:** Com as informações a respeito das ligações entre

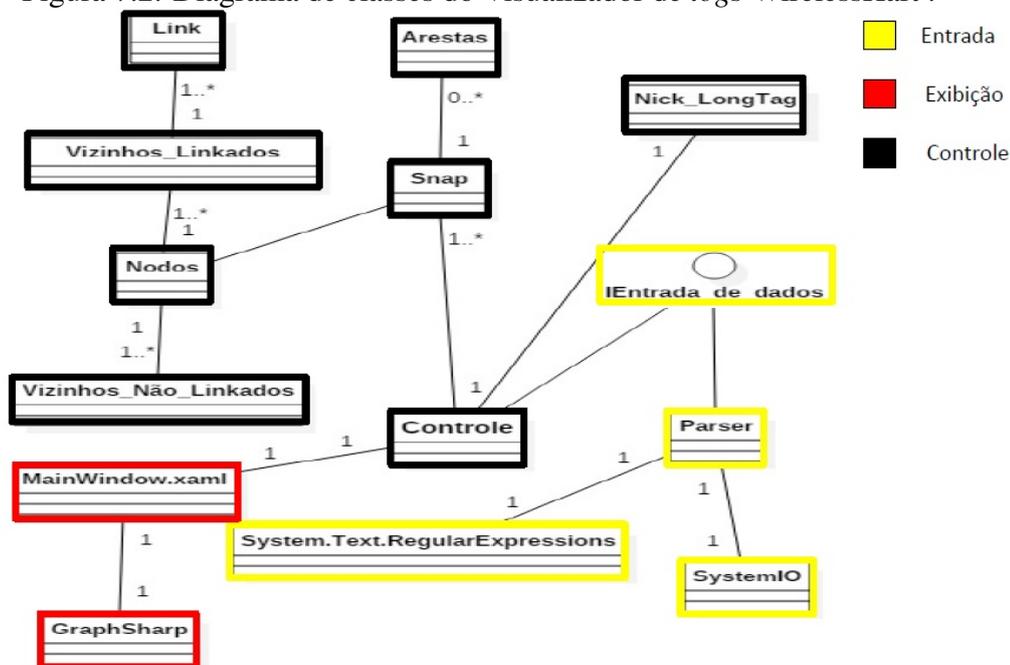
dispositivos, a ferramenta é capaz de representar a topologia da rede em forma de grafo;

- **Analisar comunicação entre vizinhos:** na rede WH são considerados vizinhos todos os outros dispositivos que se encontram dentro do raio de percepção de um determinado dispositivo. Esses vizinhos são divididos em linkados ou não linkados. Linkados são aqueles que possuem um ou mais links de comunicação direta com um dispositivo. Já os não linkados apenas estão dentro do raio de percepção do dispositivo. Na representação da topologia da rede as arestas são usadas para demonstrar a qualidade da comunicação entre os dispositivos conectados. A qualidade é representada utilizando cores que denotam a taxa de perdas nas transmissões entre os dispositivos.

7.2 Diagrama de Classes

Após definida as funcionalidades disponíveis ao usuário, o projeto segue definindo a estrutura interna do software. Para isso é utilizado o diagrama de classes que descreve os componentes de software utilizados. A imagem 7.2 mostra o diagrama de classes.

Figura 7.2: Diagrama de classes do Visualizador de logs WirelessHart .



A ferramenta pode ser entendida como três grandes módulos: entrada, controle-estruturação e exibição dos dados.

Classes do módulo de entrada de dados: *Parser*, *SystemIO*, *System.Text.Regular.Expressions*.

- **Parser:** principal classe responsável pela entrada dos dados, é conectada ao controle por meio de uma interface. Essa estratégia foi tomada para agregar modularidade ao projeto. A interface define quais serão os formatos de entrada e métodos aceitos pela classe controle. Assim caso se queira trocar a forma de entrada dos dados basta apenas substituir a classe *Parser* pelo o novo módulo sem interferir no restante da aplicação;
- **SystemIO:** classe do *framework* .Net responsável pelas leituras e controle de fluxos de arquivos;
- **System.Text.Regular.Expressions:** também pertencente ao *framework* .Net, permite o uso de expressões regulares para encontrar padrões de caracteres em arquivos. O uso de expressões regulares apresenta-se como uma boa estratégia, uma vez que os arquivos de *log* são estruturados pelo *Wi-Analys* em uma forma bem definida.

Classes do módulo de controle-estruturação: *Controle*, *Snap*, *Nick_LongTag*, *Nodos*, *Arestas*, *Vizinhos_Linkados*, *Vizinhos_Não_Linkados*, *Links*.

- **Controle:** principal classe do software, responsável por agregar outras classes, montar e controlar estruturas que guardam as informações dos *Logs*.
- **Snap:** responsável por guardar todas as informações da rede, agregando as classes *Arestas* e *Nodos*. Pode ser entendida como a estrutura que representa a fotografia do estado da rede.
- **Nodos:** essa classe representa todos os dispositivos na rede, agrega duas outras classes: *Vizinhos_Linkados* e *Vizinhos_Não_Linkados*.
- **Arestas:** classe que guarda as relações entre os nodos e informações de qualidade de comunicação.
- **Vizinhos_Linkados:** estrutura responsável por referenciar os dispositivos que possuem links estabelecidos com determinado dispositivo.
- **Link:** classe agregada à *Vizinhos_Linkados*, guarda as informações pertinentes ao link, como por exemplo : *superframe* e *time slot*.
- **Vizinhos_Não_Linkados:** também agregada à classe *Nodos*, mantém a informação de intensidade de sinal dos vizinhos captados por um determinado dispositivo.

Classes do módulo de exibição: *MainWindow.xaml*, *GraphSharp*.

- **MainWindow.xaml:** responsável por lidar com os eventos gerados pela interface e

chamar os métodos da classe controle.

- **GraphSharp**: classe responsável por construir o grafo que representa a topologia da rede.

7.3 Diagramas de sequência

Tendo os requisitos de funcionalidades e estrutura de dados definidos, a modelagem avança para a fase que guia o programador na construção das lógicas que realizam os usos destacados no diagrama de casos de uso.

Foram elaborados os seguintes diagramas a fim de abordar as seguintes funcionalidades:

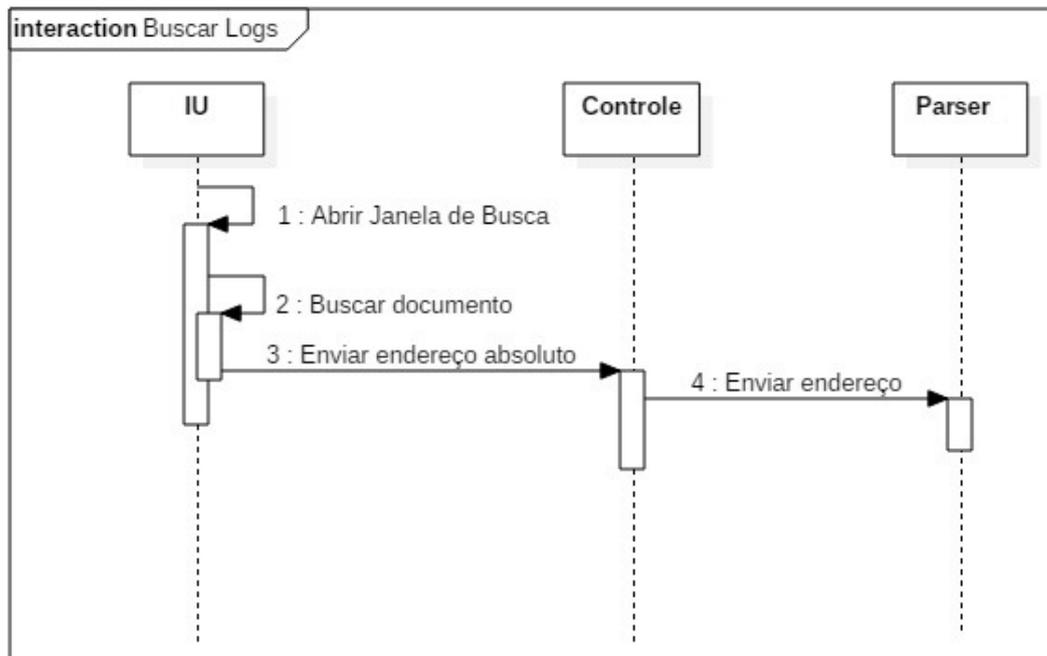
- Buscar *Logs*.
- Obter dados de Arquivo.
- Inserir Plano de Fundo.
- Montar estruturas.
- Avançar *Snap*/ Retornar *Snap*.

7.3.1 Buscar logs

O diagrama mostrado na figura 7.3 descreve a primeira funcionalidade a ser utilizada. A entidade IU, corresponde a interface de usuário, e nela são disparados os métodos de busca do arquivo que contém os dados coletados pela ferramenta Wi-Analys. Em seguida o caminho do arquivo é passado para a classe controle, que por sua vez submete a classe parser. Essa última responsável por percorrer o arquivo e capturar as informações necessárias.

7.3.2 Obter dados de Arquivo

Conhecendo o caminho de acesso ao arquivo, inicia-se a busca das informações pertinentes e a montagens das estruturas de dados. O processo de montagem das estruturas será detalhado em outro diagrama de sequência. Com o sinal para o início da montagem das estruturas a classe Controle cria uma nova *thead* para o processamento dessa tarefa,

Figura 7.3: Diagrama de sequência- Buscar logs *WirelessHart* .

o novo fluxo é necessário para que a interface de usuário não fique bloqueada. Além de reduzir a usabilidade, um bloqueio na interface pode gerar ações do sistema operacional que tenta finalizar tarefas que possuem um comportamento anômalo.

Já na classe Parser, o arquivo é percorrido linha a linha, nessa etapa já são desconsiderados os pacotes de *advertisement* executando o método Capturar Pacotes de Comandos. Os comandos bem como os dados neles expressos são colhidos em estruturas que serão enviadas para processamento na classe Controle. O *loop* de leitura segue até que seja encontrada a marcação de fim de arquivo.

7.3.3 Inserir Plano de Fundo

Assim como na busca de *log*, o arquivo da imagem a ser inserida é escolhido utilizando uma janela específica, seu caminho então é enviado para a classe controle que o imprime na porção da interface de usuário reservada para este fim. Neste caso, ao contrário da leitura do arquivo *log*, não faz-se necessário o uso de uma nova *thread* pois os tempos não são longos o suficientes para gerar um bloqueio da interface de usuário.

Figura 7.4: Diagrama de sequência-Obter dados de Arquivo.

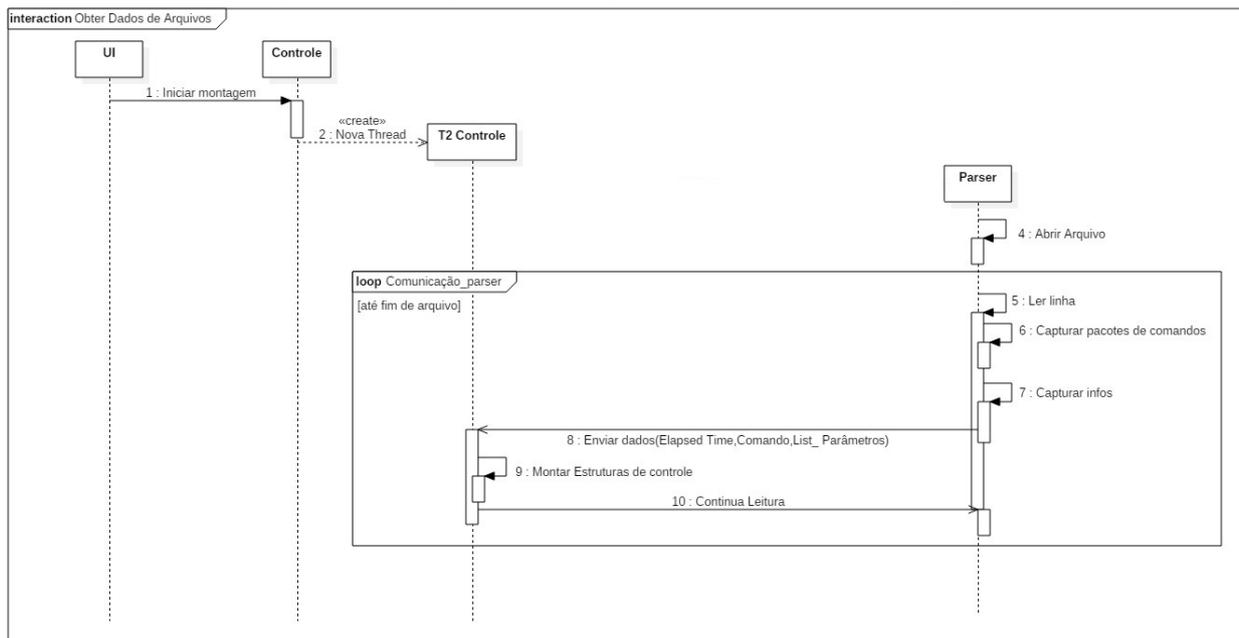
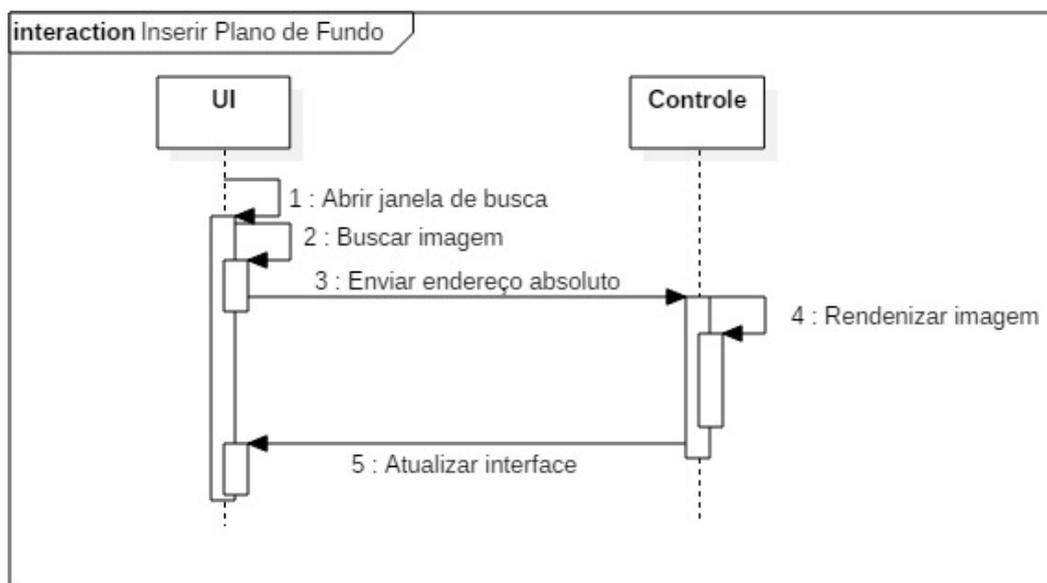


Figura 7.5: Diagrama de sequência-Inserir Plano de Fundo.



7.3.4 Montar estruturas

Principal operação da classe Controle, responsável pela construção das estruturas de dados que guardarão os dados extraídos do arquivo *log*. Os dados que são recebidos em estruturas genéricas são separados de acordo com o número do comando WH que os encapsulam. Para facilitar a visualização dos diagramas os mesmos foram divididos, cada um descrevendo o comportamento de um conjunto de dados. São eles:

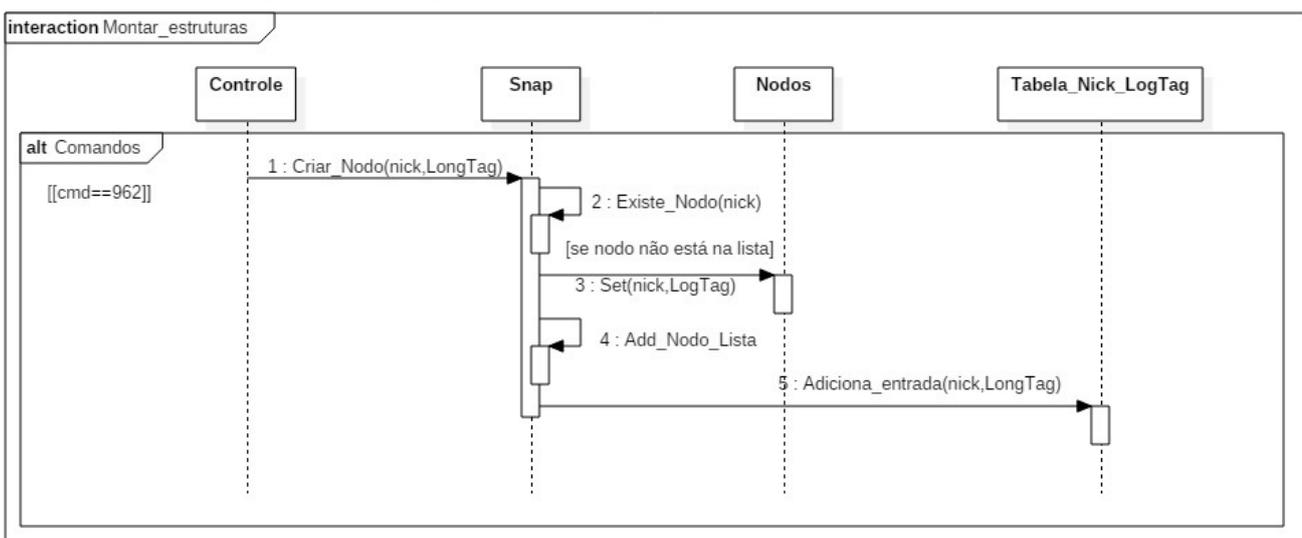
- *Write Device Nickname Address* (962);
- *Write Link*(968);
- *Delete Link* (967);
- *Report Neighbor Signal Levels* (787);
- *Report Neighbor Heath List* (780).

7.3.4.1 *Write Device Nickname Address* (962)

Quando um novo dispositivo entra na rede o gerenciador de rede atribui a ele um número, chamado *Nickname*. Esse número tem a função de identificar os dispositivos de maneira mais compacta. Por ser um dos primeiros comandos a serem executados no ingresso à rede de um dispositivo, o comando 962 (*Write Device Nickname Address*) foi utilizado para disparar os métodos que adicionam a estrutura que armazenará os dados desse novo dispositivo.

Recebidos os dados vindos da classe *Parser*, a classe *Controle* chama o método responsável por incluir um novo nodo nas estruturas internas da classe *Snap*. O novo nodo então é adicionada após passar por testes que evitam estruturas duplicas. Por último os identificadores vão para uma estrutura que relaciona o *nickname* recebido ao seu respectivo *LongTag*.

Figura 7.6: Diagrama de seqüência-Montar estruturas,comando 962.



7.3.4.2 Write Link (968) e Delete Link (967).

Após entrarem na rede, os dispositivos estabelecem links unidirecionais com o gerenciador de rede e entre si. Esses links possuem como caracterizadores tais como em qual *superframe* e *time slot* estão situados. Os comandos 968 e 967 são análogos sendo o primeiro para adicionar e o segundo para remover os links entre dispositivos. A diferença entre os dois comandos é que no momento da exclusão somente são necessárias as propriedades que caracterizam o link de forma única.

Como mostrado no diagrama de classe os nodos possuem uma estrutura que representa os dispositivos vizinhos com os quais possuem um link de comunicação. Essas estruturas guardam as informações dos links entre origem e destino. Após a inclusão ou exclusão do link na classe *Vizinhos_Linkados* a devida atualização da classe *Aresta* deve ser realizada.

Figura 7.7: Diagrama de sequência-Montar estruturas, comando 968.

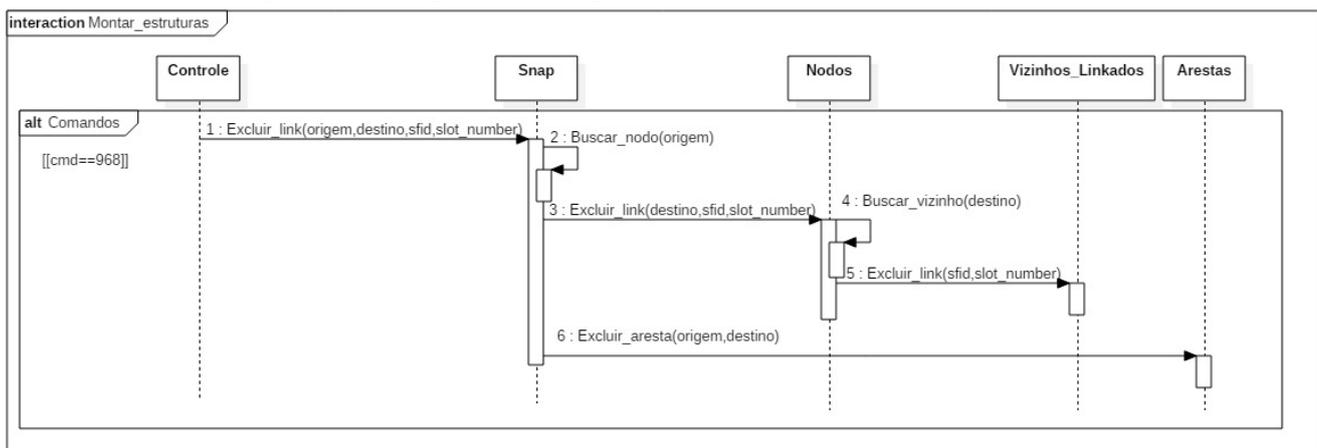
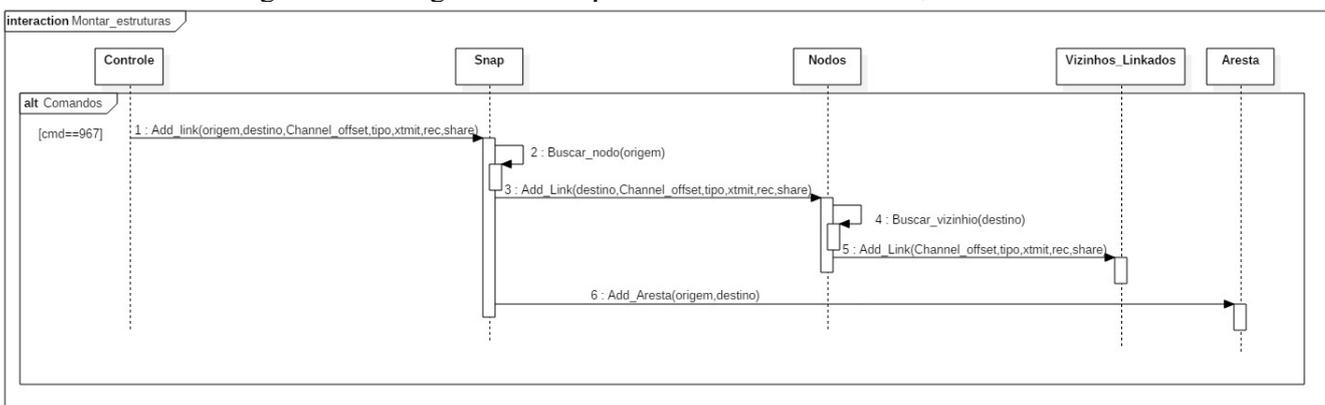


Figura 7.8: Diagrama de sequência-Montar estruturas, comando 967.

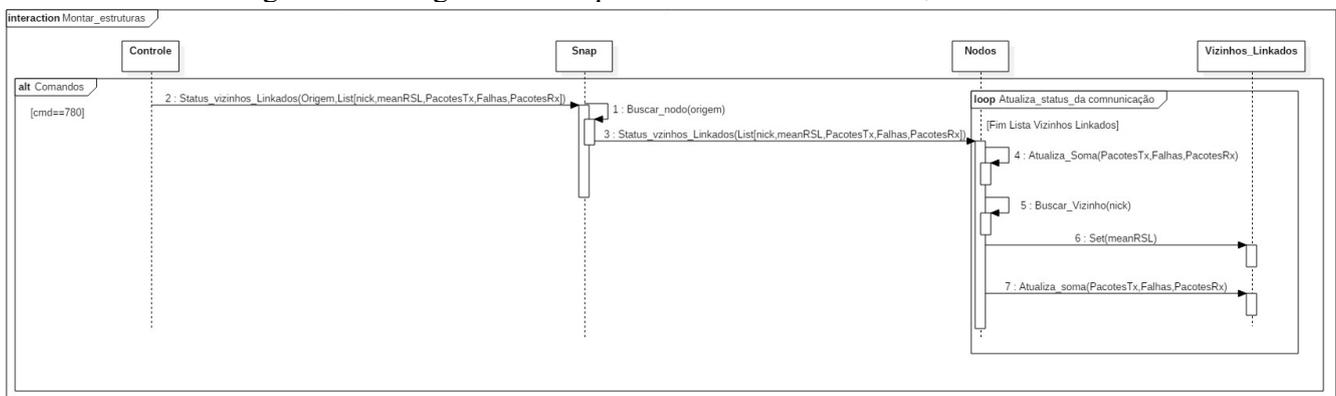


7.3.4.3 Report Neighbor Heath List (780)

As informações de qualidade entre vizinhos que possuem um link de comunicação são capturadas dos dados encapsulados no comando 780 *Report Neighbor Heath List*. Esses dados contém uma lista de todos os dispositivos linkados, e em cada elemento desta lista estão descritos o identificador *nickname*, *meanRSL*, números de pacotes enviados, pacotes recebidos e pacotes que tiveram falha de envio.

Após a devida identificação dos nodos inicia-se a fase de atualização dos dados. Esse laço de atualização continua até que toda lista tenha sido percorrida.

Figura 7.9: Diagrama de sequência-Montar estruturas, comando 780.



7.3.4.4 Report Neighbor Signal Levels (787)

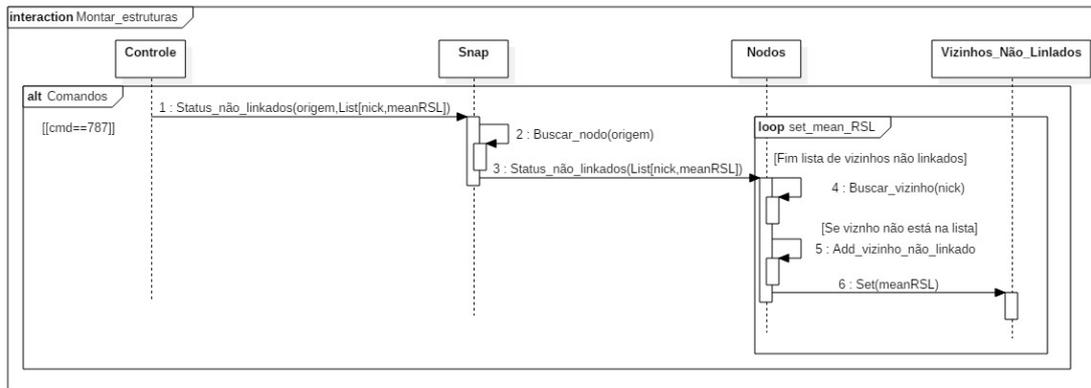
Assim como no caso do comando 780 o comando 787 também traz os dados em forma de lista, porém entre os vizinhos que não possuem links de comunicação. Após o processo de identificação dos nodos inicia-se o laço de atualização do *meanRSL* no respectivo vizinho. Caso o vizinho esteja na lista de vizinhos não linkados o mesmo é adicionado.

7.3.5 Avançar Snap/ Retornar Snap

As operações Avançar *Snap* e Retornar *Snap* possuem a mesma função: percorrer os diferentes estados da rede. Estes estados são representados pela topologia da rede e por estatísticas das conexões entre dispositivos e balizados por um período de tempo.

Por simplicidade será descrito somente o caso avançar *Snap*. A diferença entre os dois casos é somente o incremento ou decremento do valor de controle *SnapFrame_atual*.

Figura 7.10: Diagrama de seqüência-Montar estruturas, comando 787.



A cada mudança de *Snap* o grafo anterior é excluído e criado um novo a partir das estruturas *Nodos* e *Arestas*. Para isso a operação conta com dois laços que percorrem essas estruturas adicionando cada elemento ao novo grafo. Com todos elementos adicionados, a interface é atualizada a fim de exibir o novo *Snap*.

Figura 7.11: Diagrama de seqüência-Avançar *Snap*.

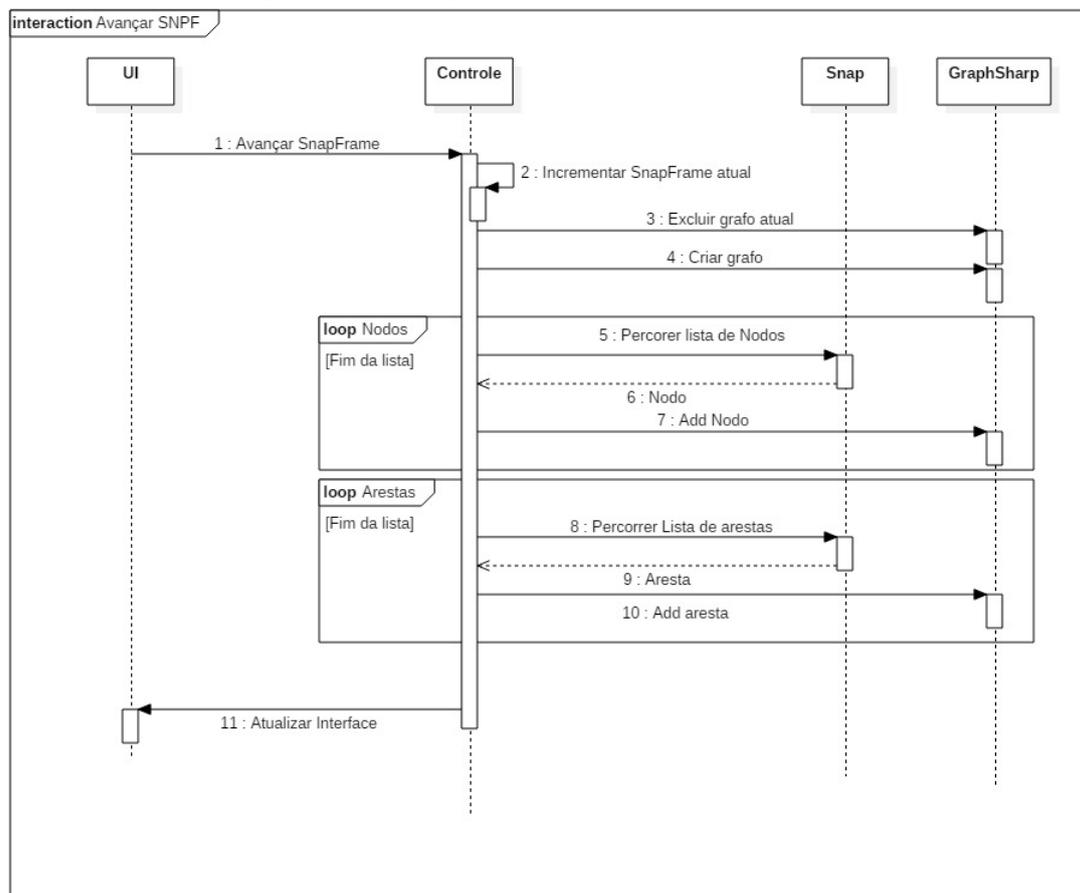
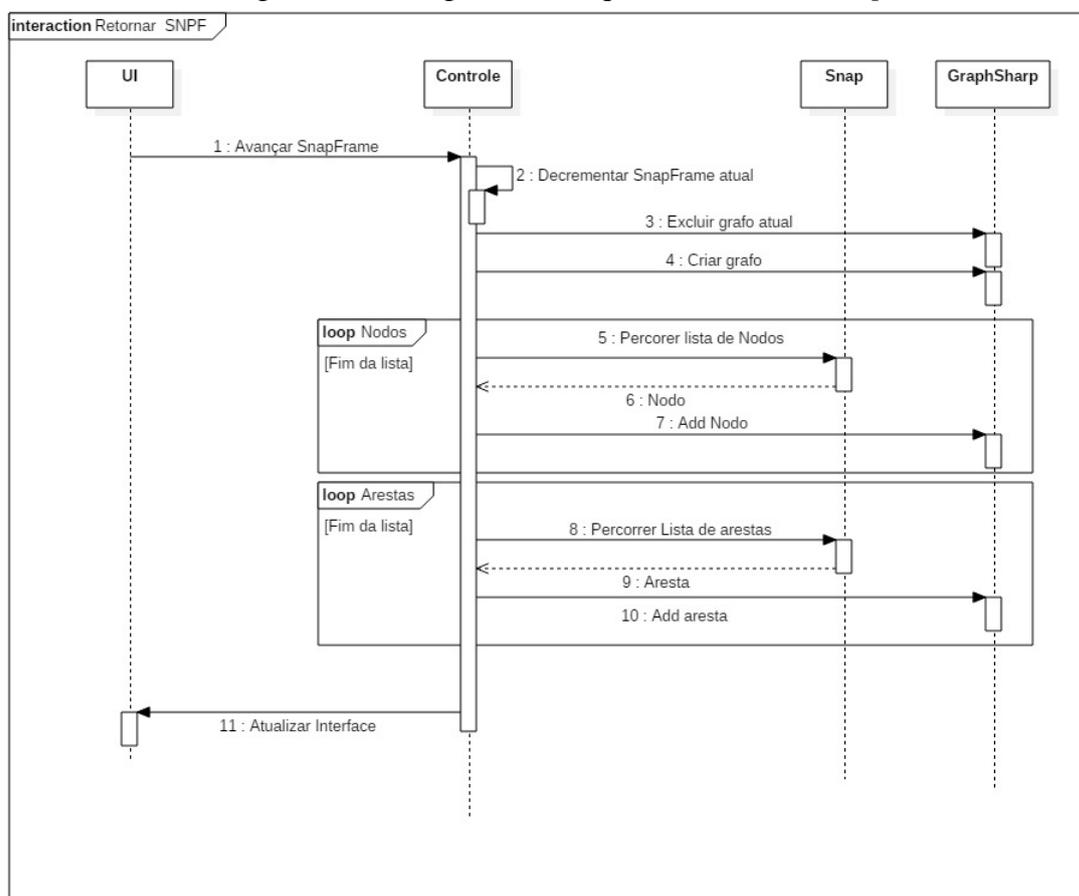


Figura 7.12: Diagrama de sequência-Retornar *Snap*.



8 RESULTADOS OBTIDOS

Para apresentar os resultados do projeto esta seção será dividida em duas seções:

- Apresentação e aspectos gerais;
- Utilização.

8.1 Apresentação e aspectos gerais.

O resultado da implementação do projeto mostrado no capítulo 7 é discutida ao longo deste seção.

Seguindo o objetivo do projeto de simplificar o processo de análise do *log Wi-Analys*, a interface de usuário foi construída. Visando o uso intuitivo, poucos menus e botões foram implementados, com esses comandos o analista pode percorrer os diversos segmentos do *log* e visualizar as informações. No anexo B pode ser vista a interface de usuário completa.

A apresentação da interface foi dividida em três partes:

- Controles;
- Informações de detalhadas;
- Topologia.

8.1.1 Controles

A figura 8.1 mostra os controles da interface de usuário.

Figura 8.1: Controles da ferramenta de visualização de *logs Wi-Analys*

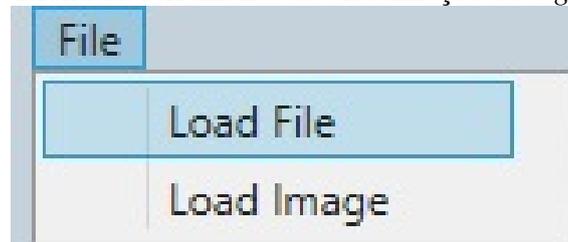


O primeiro controle a ser utilizado pelo analista deve ser a caixa de entrada de texto *Snap Time*. Nesse controle deve ser inserido o valor em minutos que define o período coberto por um *Snap*. Por exemplo: utilizando o valor mostrado na figura, o *log* será subdividido em períodos de 30 minutos e as informações pertinentes contidas nessa seção serão estruturadas pela classe *Snap*.

Seguindo, deve-se buscar os arquivos que serão utilizados na análise utilizando o

menu mostrado na figura 8.2. Em *Load File* é buscado o arquivo *log* e em *Load Image* a imagem a ser utilizada como plano de fundo para a topologia da rede. Em ambos os casos é aberta uma janela auxiliar para percorrer os diretórios e encontrar o arquivo desejado.

Figura 8.2: Menu da ferramenta de visualização de *logs Wi-Analys*



Com o arquivo devidamente carregado, pode-se navegar pelos *Snap*s utilizando os botões de setas. O botão seta para direita avança para o próximo *Snap* e o botão seta para esquerda retorna para o *Snap* anterior. Para melhor situar temporalmente o usuário são mostradas as informações: *current snap* e *test time*. A primeira indica qual *Snap* está sendo exibido no momento e a segunda mostra em minutos quanto tempo do teste capturado já foi percorrido.

8.1.2 Informações de detalhadas

Ao selecionar um dispositivo na topologia, as informações detalhadas são exibidas em um painel à direita da topologia. A imagem 8.3 mostra um exemplo de informações detalhadas de um dispositivo.

As informações iniciais identificam o dispositivo selecionado:

- **Name:** identifica o dispositivo utilizando o *long_tag*, nome configurado previamente no dispositivo;
- **Nick:** identifica o dispositivo utilizando o *nickname*, valor numérico atribuído ao dispositivo pelo gerenciador de rede;
- Topologia.

A segunda seção do painel, *Info Links*, mostra todos os links de comunicação que o dispositivo possui. Os dados a respeito dos links são dispostos nas seguintes colunas:

- **To:** indica o nickname do dispositivo com o qual o link é estabelecido. O nickname 65535 indica um endereço de broadcast;
- **SF_ID:** informa o número do superframe em que o link está inserido;

Figura 8.3: Painel de informações detalhadas da ferramenta de visualização de *logs Wi-Analys*

Name: EAWARE-02							
Nick: 3							
Info Links							
To	SF_ID	Slot	offset_Chan	Type	Xmit	Recv	Share
2	1	31	7	Broadcast	False	True	False
2	0	796	4	Normal	True	False	False
2	0	288	0	Normal	True	False	False
2	0	551	11	Normal	True	False	False
65535	0	1	0	Discovery	True	True	False
65535	4	34	13	Broadcast	True	False	False
65535	1	43	3	Join	True	False	False
65535	0	448	6	Join	False	True	False
5	0	523	0	Normal	False	True	False

Linked Neighbors				
Neighbor	Transmissions	Recv	Failures	MeanRSL
EAWARE-01	628	2362	22	-67
EAWARE-05	149	940	0	-77

Unlinked Neighbors	
Manager---	Mean RSL: -12
EAWARE-04---	Mean RSL: -1

Total Communication Values	
Transmissions:	1249
Received:	4187
Failures:	358

- **Offset_Chan:** este valor é utilizado no salto de canais, como base para calcular qual o próximo canal a ser utilizado;
- **Type:** indica o tipo do link, podem ser Normal, Discovery, Broadcast ou Join;
- **Xmit:** valor verdadeiro ou falso que indica se o link é de transmissão;
- **Recv:** valor verdadeiro ou falso que indica se o link é de recepção;
- **Share:** valor verdadeiro ou falso que indica se o link é compartilhado.

A terceira seção do painel, *Linked Neighbors* mostra uma lista de vizinhos que possuem *link* de comunicação com o dispositivo. Para cada vizinho *linkado* são dispostas

as informações a respeito da comunicação:

- **Tranmissions:** indica o número de pacotes transmitidos para o vizinho *linkado*;
- **Recv:** indica o número de pacotes recebidos do vizinho *linkado*;
- **Failures:** indica o número de pacotes transmitidos que não chegaram ao destino corretamente;
- **MeanRSL:** Indica o valor de RSL médio percebido pelo dispositivo.

A quarta seção do painel, *Unlinked Neighbors* mostra a lista de vizinhos não linkados. Além dos identificadores dos vizinhos são mostrados os valores médios de RSL, de cada vizinho não *linkado*, percebidos pelo dispositivo.

A quinta seção, *Total Communications Values* mostra os dados de comunicação do dispositivo. Os valores de pacotes transmitidos, pacotes recebidos e falhas de transmissão de todos os vizinhos linkados são somados e exibidos nessa seção.

8.1.3 Topologia

A topologia da rede é apresentada na forma de grafo de arestas coloridas e direcionadas, como o apresentado na figura 8.4.

A direção da seta indica o sentido da comunicação. Utilizando a topologia mostrada na imagem como exemplo, a aresta entre *Eawere-05* e *Eawere-01* mostra que existe links de transmissão apenas no sentido do primeiro citado para o segundo.

As cores indicam a qualidade da comunicação ente os dispositivos. O fator utilizado para classificar a qualidade foi a proporção entre pacotes transmitidos e número de falhas de transmissão.

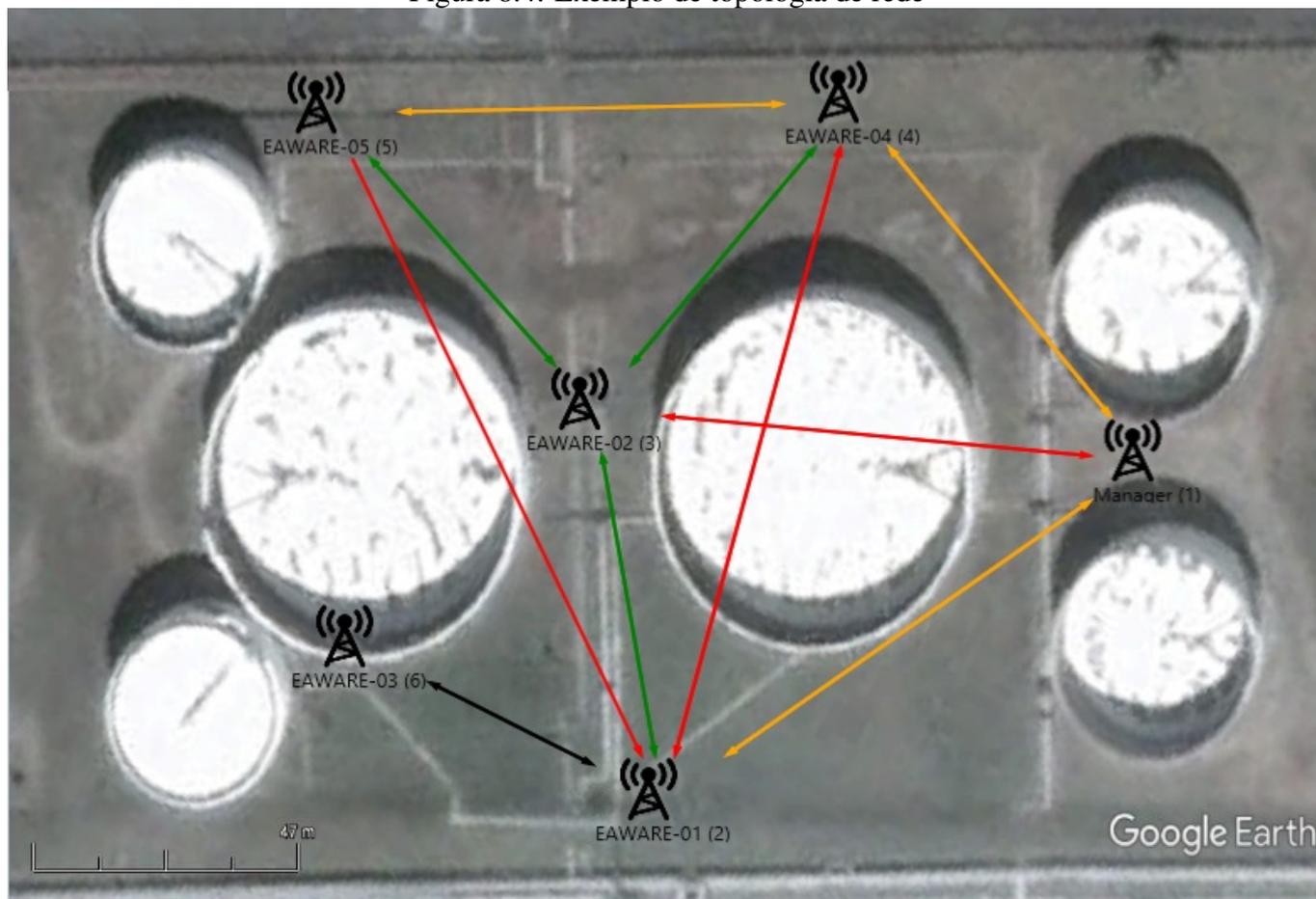
Foram estabelecidos três níveis para a qualidade da comunicação: bom, intermediário, ruim. Cada nível é mapeado para a topologia como uma cor, sendo:

- **Verde** - bom;
- **Laranja**- intermediário;
- **Vermelho**- ruim.

Para as arestas simples, ou seja, em que existe apenas um sentido de transmissão o cálculo da taxa de falhas é calculado da seguinte forma:

$$Prop_{falhas} = \frac{dispositivo.falhas * 100}{dispositivo.trasmit}$$

Figura 8.4: Exemplo de topologia de rede



Já nos casos em que existe transmissão nos dois sentidos é utilizado o seguinte cálculo:

$$Prop_{falhas} = \frac{(dispositivo.falhas + vizinho.falhas) * 100}{dispositivo.transmit + vizinho.transmit}$$

Por realizar uma média entre os dados de dois dispositivos, alguns comportamentos podem ser mascarados. Como, por exemplo, no caso em que um sentido da conexão possui boa qualidade e no sentido inverso uma qualidade ruim. No caso em que a magnitude dos valores do sentido com boa qualidade é muito maior que do sentido inverso será apresentado um valor de boa qualidade total. O usuário deve então recorrer aos valores detalhados descritos no campo *Linked Neighbors* para encontrar as informações detalhadas.

Em ambos os casos os valores arbitrários de qualidade são classificados de acordo com a tabela 8.1:

As arestas que em um determinado *snap* ainda não possuem valores de qualidade determinados são permanecem como a cor preta até que os valores de transmissões e

Tabela 8.1: Classificação de cores de acordo com a proporção de falhas

Qualidade	Proporção
Boa	Menor que 25 %
Intermediária	Maior que 25 %
Intermediária	Menor que 50%
Ruim	Maior que 50%

falhas seja conhecido.

8.2 Utilização

Neste capítulo serão demonstrados usos possíveis da ferramenta e uma discussão sobre os avanços trazidos para análise em comparação com o método utilizado atualmente.

Os dados utilizados nos cenários abaixo foram coletados utilizando a ferramenta *Wi-analys* em teste realizado em um ambiente real e não controlado.

8.2.1 Cenário 1

O cenário 1 mostra a situação da formação inicial da rede, onde interesse principal é conhecer a distribuição dos links. Os dispositivos começam a ser inseridos após aproximadamente doze minutos de iniciada a captura. A imagem 8.5 mostra o fragmento do arquivo *log* correspondente a essa situação. As primeiras linhas são referentes ao processo de negociação de entrada do dispositivo, no qual comandos de identificação, negociação de chaves de sessão e atribuição do identificador *nickname* são executados. Após a entrada efetiva do dispositivo são distribuídos os links com os quais o dispositivo irá realizar a comunicação com vizinhos ou descoberta de novos vizinhos. Nessa primeira distribuição são escritos 8 links, não havendo ainda vizinhos além do gerenciador de rede, os endereços de destino são o próprio gerenciador ou endereço de *broadcast*.

Para analisar este cenário foi definido um tempo de *Snap* de cinco minutos. Com esse tempo definido o dispositivo e seus links são exibidos apenas no terceiro *Snap*, apresentado na figura 8.6.

8.2.2 Cenário 2

No segundo cenário a rede já possui outros dispositivos e diversos links entre os membros da rede foram definidos. A figura 8.7 mostra o trecho do *log* em que são enviadas ao gerenciador de rede os primeiros relatórios de qualidade das comunicações.

Por não possuir link direto com o gerenciador de rede, o dispositivo *EAWARE-2* envia seus dados para o gerenciador por intermédio do dispositivo *EAWERE-01*.

O resultado da visualização desse trecho de log é mostrado nas figuras 8.8 e 8.9.

As figuras mostram o correto funcionamento da codificação das cores quanto a proporção de falhas por transmissão, bem como a capacidade de capturar as informações mesmo quando transmitidas de forma indireta para o gerenciador de rede.

Os dois casos apresentados, além de demonstrarem o funcionamento correto da ferramenta, têm como função validar o objetivo do projeto. Mesmo em comparação com um fragmento do arquivo *log*, organizar de forma simples e clara aumenta a percepção global do analista, aumentando assim a eficiência da análise.

Figura 8.6: Visualização distribuição de links-Cenário 1



Figura 8.8: Visualização dados de qualidade *EAWERE-01* - Cenário 2



Figura 8.9: Visualização dados de qualidade *EAWERE-02* - Cenário 2



9 CONCLUSÃO

A aceitação do protocolo *WirelessHART* por parte das indústrias que necessitam de segurança e confiabilidade, passa por um conjunto de ferramentas que sejam capazes de testar tanto o desenvolvimento novos dispositivos quanto projetos de novas redes WH. As ferramentas disponíveis, tanto comercialmente quanto na área acadêmica, suprem parcialmente essa necessidade. A complexidade das soluções disponíveis pode induzir a erros na análise, além do desperdício de tempo com o processamento manual de arquivos de saída. A ferramenta proposta é um passo em direção ao ganho de eficiência no processo de análise de redes WH.

Ao fim do projeto pode-se constatar a validade da proposta deste trabalho. A ferramenta desenvolvida como forma de validação de conceito atendeu as expectativas iniciais. Os resultados obtidos demonstraram o correto funcionamento da ferramenta bem como o ganho de percepção das informações guardadas no arquivo *log*.

Apesar dos resultados, são necessárias algumas consistências de software. Como por exemplo, o uso de janelas de tamanho ajustável. Como trabalho futuro de incremento na ferramenta pode-se citar: a geração de gráficos que expressem as mudanças dos valores de transmissão, falha e recepção de cada dispositivo ao longo do tempo e a introdução de um campo gráfico para exibir a ocupação dos *superframes* de forma mais intuitiva.

REFERÊNCIAS

- CHEN, D.; NIXON, M.; MOK, A. **Why WirelessHART**. [S.l.]: Springer, 2010.
- FERRARI, P. et al. On the implementation and performance assessment of a wirelesshart distributed packet analyzer. **IEEE Transactions on Instrumentation and Measurement**, IEEE, v. 59, n. 5, p. 1342–1352, 2010.
- GROUP, F. **WirelessHART Overview**. 2014. Available from Internet: <[http://http://en.hartcomm.org/hcp/tech/wihart/wireless_overview.html](http://en.hartcomm.org/hcp/tech/wihart/wireless_overview.html)>.
- KUNZEL, G. **Ambiente para avaliação de estratégias de roteamento para redes WirelessHART**. Dissertação de Mestrado.UFRGS, 2012.
- LEAL, A. B. **Ferramenta para análise de logs de equipamentos de teste de conformidade, usados com o protocolo WirelessHART**. Trabalho de Conclusão de Curso. UFRGS, 2016.
- LORENÇATO, A. d. A. **Analisador de redes wirelesshart**. Dissertação de Mestrado.UFRGS, 2013.
- MACHADO, T. M. **Analisador de redes WirelessHART com capacidade de detecção de coexistência**. Dissertação de Mestrado.UFRGS, 2014.
- RUMBAUGH, J.; JACOBSON, I.; BOOCH, G. **Unified Modeling Language Reference Manual, The**. [S.l.]: Pearson Higher Education, 2004.
- SPEC-065. **2.4GHz DSSS O-QPSK Physical Layer Specification**. [S.l.], 2007.
- SPEC-075. **TDMA Data Link Layer Specification**. [S.l.], 2008.
- SPEC-085. **Network Management Specification**. [S.l.], 2009.
- SPEC-127. **Universal Command Specification**. [S.l.], 2008.
- SPEC-151. **Common Practice Command Specification**. [S.l.], 2008.
- SPEC-155. **Wireless Command Specification**. [S.l.], 2008.

ANEXO A — TRABALHO DE GRADUAÇÃO - I

- Título: Análise de *logs* de dados de diagnóstico de redes *WirelessHART*
- Data: Maio de 2016

Análise de logs de dados de diagnóstico de redes *WirelessHART*

Matheus Maia de Souza¹,
João César Netto¹

¹Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Av. Bento Gonçalves, 9500 - Porto Alegre, RS - Brasil

{mmsouza, netto}@inf.ufrgs.br

Abstract. *The use of automated systems are spreading through many areas, in industry has significant importance. The consolidate method that uses cables as communication medium imposes some restrictions, for exemple : High restructuring cost, a constant maintenance and cable changes. WirelessHART get around this restrictions offering a wireless solution. After years of research the protocol show itself a robust and reliable alternative for a challenging environment as factory floor. However, there is a need for new support tools in certain areas, for instance, in test phase. This work will be showing a WirelessHART protocol overview, its features and organization. Also will be proposed an analysis tool for data logs coming from WirelessHART networks sniffing.*

Resumo. *O uso de sistemas automatizados está espalhado por diversos setores, na indústria tem significativa relevância. O método já consolidado que faz uso de cabos como meio de comunicação impõe algumas restrições como por exemplo: Alto custo para reestruturação e cuidado constante com a manutenção e troca de cabos. O protocolo WirelessHART contorna estas restrições oferecendo uma solução sem fio. Após anos de pesquisa o protocolo WirelessHART se mostra uma alternativa robusta e confiável para ambientes tão desafiadores como o chão de fábrica. Porém, ainda existe a necessidade da criação de ferramentas de suporte em certas áreas, como por exemplo na fase de testes. Neste trabalho será apresentado um resumo da organização do protocolo e suas funcionalidades e uma proposta de ferramenta para análise de dados vindos de capturas realizadas em redes WirelessHART.*

1. Introdução

O processo de automatização industrial é essencial para o aumento da produtividade e redução de custos. Sensores e atuadores são conectados a redes para monitorar e gerenciar os dispositivos. A solução clássica são redes que utilizam cabos para transmitir as informações, medida a qual torna mais cara e menos flexível a implementação de redes de comunicação para ambientes industriais. Com o surgimento do protocolo *WirelessHART* (WH), primeira tecnologia de comunicação sem fio certificada para uso industrial, muitas vantagens em relação aos tradicionais sistemas cabeados surgiram: Redução de custos com infraestrutura, possibilidade de instalação em locais de difícil acesso, redução do tempo de instalação, entre outros. Entretanto, tal tecnologia ainda carece de estudos para

alcançar maior aceitação no mercado. Com a intenção de preencher esta lacuna será apresentado neste trabalho o projeto de software para análise de dados obtidos em testes de redes WH.

Para contextualizar o leitor será apresentado na seção 2 uma visão geral do protocolo, mostrando suas principais características e funcionalidades. Em seguida na seção 3, será apresentada a ferramenta *sniffer* utilizada para monitorar a rede *WirelessHART*. Ainda nesta seção serão relatados os desafios relacionados ao formato e volume de dados gerados. Na seção 4 serão relatados trabalhos no mesmo campo de estudo, nesta seção ainda será traçado um paralelo entre os estudos apresentados e este trabalho. A seção 5 descreve como maior detalhamento o problema a ser resolvido bem como quais técnicas e estratégias serão utilizadas. Por fim, na seção 6 um planejamento e considerações finais serão apresentadas.

2. *WirelessHART*: Uma visão geral

O protocolo de comunicação *HART protocol* é consolidado no setor de automação industrial com mais de trinta milhões de dispositivos instalados, a fim de estender as capacidades do seu protocolo a *HART Communication Foundation* na revisão de número sete introduziu um novo protocolo voltado para a comunicação sem fio: O *WirelessHART*. Assim como nas revisões anteriores, a revisão número sete tem como objetivo o controle e supervisão de sensores e atuadores. Além das capacidades próprias das redes sem fio, pode-se explorar a compatibilidade com redes *HART* já instaladas, com o uso de adaptadores para comunicação sem fio, para integrar os recursos já existentes em novo modelo para a planta.

Para organização da sua arquitetura o protocolo *WirelessHART* utiliza o modelo de camadas OSI (*Open Systems Interconnection*) [Chen et al. 2010]. A figura 1 mostra a organização das camadas do protocolo com uma breve descrição da funcionalidade de cada camada.

A seguir cada camada e outras características do protocolo serão discutidas.

2.1. Camada Física

A camada física tem como objetivo: Fornecer serviços para a camada de dados e controlar a interação entre dispositivo e meio de transmissão. Para isto utiliza parâmetros bem definidos nos documentos da norma [SPEC-065 2007]. Para o tráfego das informações é utilizada a faixa de frequência reservada ISM 2.4 GHz. [SPEC-065 2007]

- A camada física usa um conjunto reduzido de funcionalidades e características do padrão IEEE802.15.4 definido em IEEE 802.15.4-2006 Section 6. Este subset define algumas regras que devem ser seguidas por todas implementações do protocolo *WirelessHART*.
- Em uma rede *WirelessHART* a menor janela de tempo para uma comunicação é 10 ms. Quando em modo *unicast* a mensagem de requisição e de confirmação devem ser enviadas dentro da janela de tempo. Já em *broadcast* apenas uma mensagem é enviada dentro da janela de tempo uma vez que as mensagens de *broadcast* não recebem confirmação.
- O menor tempo entre duas mensagens é 1ms, tempo entre o fim da transmissão e o início do envio de confirmação por parte do receptor.

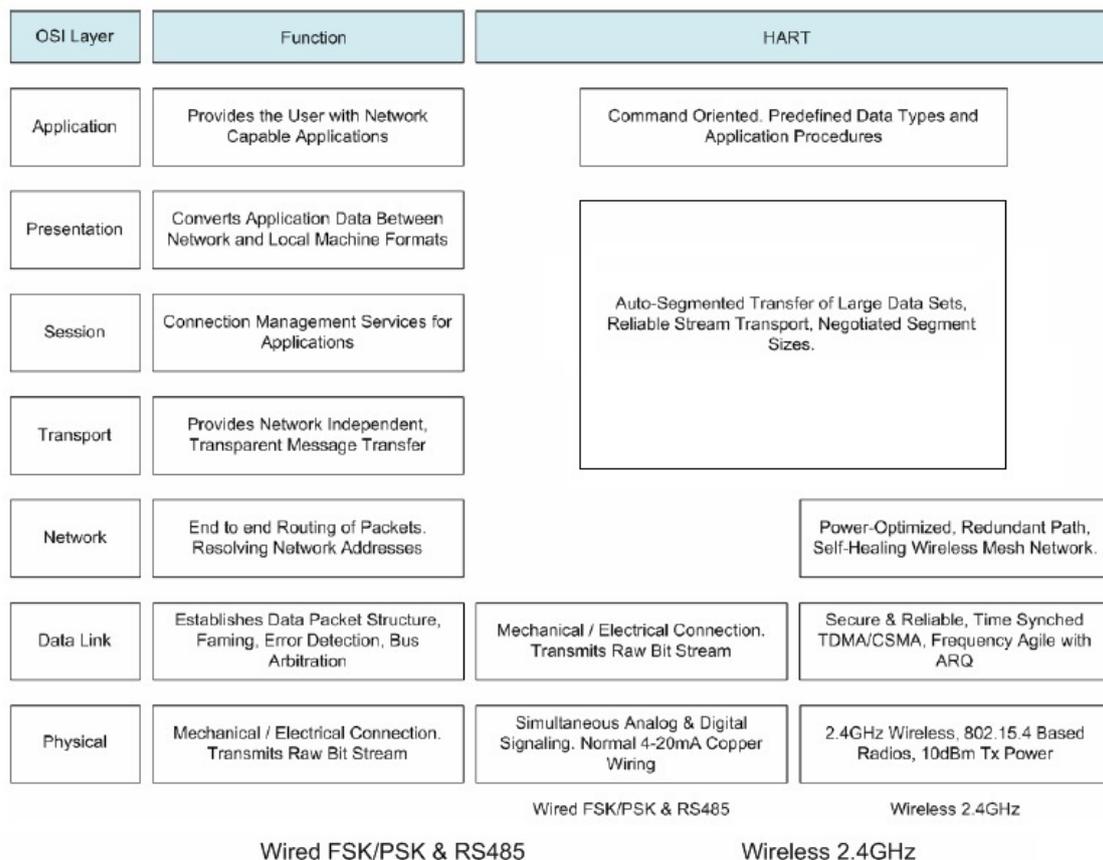


Figura 1. Modelo de camadas OSI *WirelessHART*

- Todas mensagens trocadas em uma rede *WirelessHART* utilizam o mesmo tipo de dados definidos na norma IEEE 802.15.4
- São utilizados somente os 15 primeiros canais (11-25) da faixa ISM, o canal 26 não é utilizado por ter seu uso não autorizado em muitas localidades.
- Para evitar interferências é utilizada a técnica de salto de canais a cada transmissão.
- As colisões entre transmissões simultâneas são mitigadas pela técnica de CCA que analisa o meio antes de transmitir a mensagem, caso o meio já esteja ocupado o dispositivo deve aguardar sua nova oportunidade para transmitir.
- A Sensibilidade do dispositivo é normalizada, sendo exigido para todo o sistema (rádio, circuito, conectores e etc) o valor de -85 dB ou melhor.
- Potência de transmissão: Todos dispositivos devem ter um EIRP (*Equivalent isotropically radiated power*) de +10dBm (10mW) com tolerância de +/-3dB. A faixa programável para a potência varia entre -10dBm a +10dBm.
- A seguir é apresentada uma tabela (1) com os valores esperados para o alcance em campo aberto e para ambientes fechados. Os valores são estimados assumindo: Ganho unitário em uma antena omnidirecional, intensidade de recepção de -82dBm e ambiente sem interferência.

2.2. Camada de Enlace

A camada de enlace ou *Data link layer* deve garantir a transferência dos dados entre os dispositivos da rede, para isso dispõe de recursos de detecção e correção de erros e de

Tabela 1. Valores esperados de distância vs potência.

Ambiente	Potência	Distância (em metros)
Externo (com visada)	0 dBm	100
Externo (com visada)	10 dBm	200
Interno	0 dBm	35
Interno	10 dBm	75

mecanismos de controle de acesso ao meio. O recurso de controle de acesso ao meio visa garantir que a qualidade da comunicação não seja afetada por colisões entre quadros WH. As responsabilidades são divididas em duas subcamadas: *Logical Link Control* (LLC) e *Media Access Control* (MAC) . [SPEC-075 2008]

2.2.1. Logical Link Control (LLC)

A LLC é responsável pelo controle de erros, controle de fluxo e montagem do quadro *WirelessHART*.

A figura 2 mostra estrutura do quadro da camada de dados.

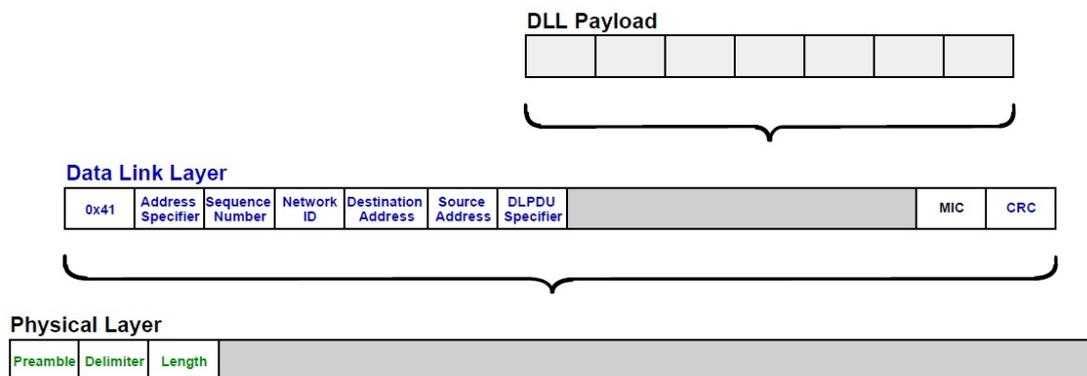


Figura 2. Data-Link Protocol Data Unit (DLPDU)

O campo *DLPDU Specifier* tem função importante no controle de fluxo do protocolo, em um byte codifica três características distintas pertencentes ao quadro WH: Prioridade, *network key* e tipo de pacote.

- A prioridade é dividida em quatro níveis: *Command*, *Process Data*, *Normal*, *Alarm*. Sendo o primeiro nível citado o mais prioritário seguido pelos demais na ordem em que foram dispostos.
- *Network Key*: Define tipo de chave criptográfica utilizada na transmissão (*network key* ou *public key*).
- Os tipos de mensagens são classificados como: *ACK*, *Advertise*, *Keep Alive*, *Disconnect*, *Data*.

No byte *DLPDU Specifier* os dois primeiros bits mais significativos são reservados para futuras aplicações.

2.2.2. Medium Access Control (MAC)

Para evitar colisões na troca de mensagens e garantir uma comunicação determinística o protocolo *WirelessHART* emprega a técnica TDMA. A principal estrutura utilizada nesta técnica é o superframe. Como mostrado na figura 3, a estrutura é composta por slots de tempo que definem o momento em que a comunicação deve ocorrer.

Os tempos de transmissão, recepção e resposta dentro dos slots de tempo também são regulados pela camada MAC.

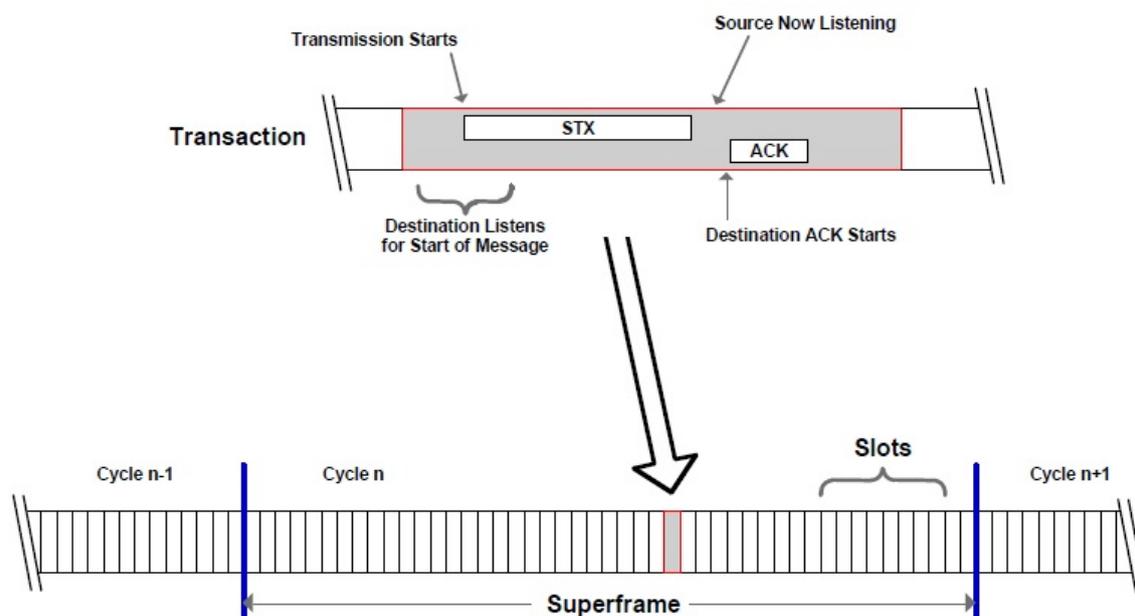


Figura 3. Superframe e Slot de tempo

A subcamada MAC tem então como objetivo:

- Manter a sincronização dos tempos de comunicação.
- Identificar os slots de tempo que devem ser utilizados.
- Propagar mensagens vindas de vizinhos.
- Repassar para a camada física quadros vindos da camada de rede.

2.3. Camada de Rede

Como mostrado na figura 1 a camada de rede do protocolo WH engloba três camadas do padrão OSI: Rede, transporte e sessão. [SPEC-085 2009]

Assim as funcionalidades das três camadas também são agrupadas na camada de rede, são elas:

- Endereçamento e entrega de dados.
- Comunicação confiável.
- Segmentação.
- Controle de erros.
- Estabelecimento da conexão entre dois nós da rede.

Os tipos de comunicação utilizados no protocolo WH são: *Request/Response* e *Broadcast*. A primeira define uma comunicação *host* e um dispositivo único. A segunda utiliza o mesmo padrão *Request/Response* porém com um endereço de destino especial chamado de *Broadcast* que envia a mensagem para todos os dispositivos capazes de enxergar a rede do remetente.

São dois os tipos de roteamento das mensagens previstos na norma: Grafo ou *Source*. O primeiro é um conjunto de links entre os dispositivos, oferecendo caminhos redundantes para que o quadro chegue ao seu destino. O caminho escolhido depende das condições da rede no momento da transmissão. O segundo tipo, é uma rota direta e estática entre dois nós da rede e é especificada no cabeçalho do quadro *WirelessHART*.

A camada de rede gerencia as comunicações fim-a-fim utilizando sessões criptografadas. Para isto, os dispositivos devem manter informações de segurança tais como chaves de encriptação. O padrão utilizado é o AES 128 bit, com dois tipos de chaves: *Join Key* e *session key*.

A *Join Key* deve ser configurada previamente para que o dispositivo esteja habilitado a entrar na rede. A *session key* é configurada após a entrada do nó na rede e previne que ataques maliciosos possam injetar dados falsos nos fluxos válidos.

2.4. Camada de Aplicação

A aplicação é a última camada da pilha de protocolos e também a mais próxima do usuário final. É na camada de aplicação que os comandos são implementados, definindo portanto o comportamento do comando e os tipos de dados que devem ser carregados nas mensagens. Cada número de comando é uma especificação única e não ambígua do dado e tamanho de um determinado quadro *WirelessHART*.

Os comando são divididos em classes:

- *Universal Commands* : São comandos que devem ser suportados por todos dispositivos HART.[SPEC-127 2008]
- *Common Practice Commands* : Comandos implementados por um grande número de dispositivos e devem ser suportados sempre que possível.[SPEC-151 2008]
- *Non-public Commands* : Comandos especiais que tem seu uso limitado para ambiente de fabricação dos dispositivos.
- *Wireless Commands*: São a base da comunicação das redes WirelessHART sendo mandatório para produtos *WirelessHART*. [SPEC-155 2008]
- *Device Family Commands*: Grupo de comandos usados para configuração de dispositivos de campo, sem a necessidade de implementações individualizadas.
- *Device-Specific Commands* : Definido pelo fabricante de acordo com o propósito de seu produto.

3. Ferramenta de captura de rede - *Wi-Analys*

Wi-Analys é um *sniffer* de rede *WirelessHART* utilizado para testes de conformidade e diagnósticos. A ferramenta é composta por um dispositivo rádio receptor conectado à uma estação de trabalho executado software de captura. O dispositivo cobre 16 canais simultaneamente e captura todas mensagens do padrão IEEE 802.15.4.

Se um dado é identificado como válido então o quadro é apresentado separando os campos em colunas e apresentado em uma interface. A figura 4 mostra a interface do software utilizado em conjunto com *sniffer*.

Packet Number	Elapsed Time	RSL	Channel	PDU	Net ID	To	From	Graph ID	Dest	Src	Payload
256855	10,292,968.439	-45	17	Data	1236	0001	A1MATWILL 0000	0000	F980	001E1X 010Z 030405	40 00 00 [Response] 0, 23, 0, FE 20, 33, 0, 41 00 00 787, 7, 0, 00
256856	10,292,968.538	-43	17	ACK	1236	A1MATWILL 0001					8F 00 00
256893	10,293,963.432	-50	13	Data	1236	A1MATWILL 0001	0000	0000	001E1X 010Z 030405	F980	961, 16, 91 962, 2, 00 963, 29, 00
256894	10,293,968.086	-39	13	ACK	1236	0001	A1MATWILL				
256923	10,294,963.482	-46	17	Data	1236	0001	A1MATWILL 0000	0000	F980	000B	CF 00 00 [Response] 961, 17, 0, 91 962, 3, 0, 00 963, 30, 0, 00
256924	10,294,967.656	-43	17	ACK	1236	A1MATWILL 0001					
256965	10,295,963.444	-51	13	Data	1236	A1MATWILL 0001	0000	000B	F980		90 00 00 965, 5, 00 967, 8, 00 967, 8, 00 974, 5, 03 969, 4, 00 971, 3, 00
256966	10,295,967.455	-39	13	ACK	1236	0001	A1MATWILL				
257039	10,296,163.460	-51	17	Data	1236	0001	A1MATWILL 0080	0000	F980	000B	D0 00 00 [Response] 965, 1, 10,] 967, 11, 0, 00 967, 11, 0, 00 974, 8, 0, 03 969, 6, 0, 00 971, 4, 0, 00
257040	10,296,167.665	-45	17	ACK	1236	A1MATWILL 0001					
257066	10,299,163.459	-45	13	Data	1236	A1MATWILL 0001	0000	000B	F980		91 00 00 963, 29, 00 974, 5, 02 969, 4, 00
257067	10,299,167.634	-39	13	ACK	1236	0001	A1MATWILL				
257104	10,300,163.469	-49	17	Data	1236	0001	A1MATWILL 0080	0000	F980	000B	D1 00 00 [Response] 963, 30, 0, 00 974, 8, 0, 02 969, 6, 0, 00
257105	10,300,167.497	-43	17	ACK	1236	A1MATWILL 0001					
257168	10,302,163.480	-50	17	Data	1236	0001	A1MATWILL 0000	0000	F980	000B	90 00 00 799, 3, 00
257169	10,302,166.183	-44	17	ACK	1236	A1MATWILL 0001					
257200	10,303,163.478	-46	13	Data	1236	A1MATWILL 0001	0000	000B	F980		D0 00 00 [Response] 799, 11, 0, 00
257201	10,303,166.500	-38	13	ACK	1236	0001	A1MATWILL				
257238	10,304,163.490	-46	17	Data	1236	0001	A1MATWILL 0080	0000	F980	000B	43 00 00 [Response] 1, 5, FE, 01
257239	10,304,166.650	-43	17	ACK	1236	A1MATWILL 0001					

Figura 4. Captura de tela *Wi-Analys*

Para que o dispositivo possa decifrar a mensagem é necessário que o mesmo conheça as chaves de criptografia. Isto faz com que seja necessário o início do sensoriamento antes que os nós se conectem à rede. Já que as chaves de sessão são trocadas quando o dispositivo entra na rede

A aplicação possui também a opção de exportar os dados em arquivos que podem ser lidos e editados por ferramentas de processamento de planilhas. A grande quantidade e complexidade dos dados dificulta a análise, sendo necessário um usuário com bom conhecimento da norma para realizar um diagnóstico da rede.

4. Trabalhos Relacionados

Esta seção apresenta trabalhos relacionados a ferramentas de avaliação, em diferentes aspectos, de redes *WirelessHART*.

Em [Lorençato 2013] o autor descreve duas ferramentas para análise de redes WH. Coletor de Dados Monocanal: Esta ferramenta é composta por duas entidades, coletor de dados e gerenciador da unidade de coleta de dados. A unidade de coleta de dados utiliza

como base um dispositivo WH, mantendo funções como a sintonização dos canais. Já o gerenciador por sua vez é responsável pela coleta e pelo processamento dos *links* de comunicação. A captura então é feita pelo coletor com base nas informações recebidas.

A segunda proposta utiliza um hardware capaz de capturar 16 canais simultaneamente, esta abordagem dispensa a necessidade da unidade de gerenciamento e de cuidados com os tempos de sincronização. Porém, após a coleta os dados devem ser processados para que sejam colocados em ordem cronológica. Os dados então são analisados e apresentados utilizando um software específico para o arquivo de saída.

Em [Machado 2014] é descrita uma ferramenta que realiza o sensoriamento da rede e apresenta estatísticas a respeito da mesma. São utilizadas duas peças de hardware, uma para capturar dados e outra para armazenar informações em um cartão de memória. Além das estatísticas a ferramenta possui a função de detecção de interferência.

Em [Kunzel 2012], o autor propõe um ambiente para avaliação de estratégias de roteamento. Os algoritmos de roteamento, de uma forma geral, buscam uma rota que minimize alguma função de custo. Como por exemplo número de saltos ou consumo de energia. Os dados capturados são enviados posteriormente para o software que analisa os registros e apresenta as informações utilizando grafos. O autor finaliza avaliando algoritmos de roteamento e demonstrando um experimento com uma rede *WirelessHART* operacional.

5. Objetivos e Métodos

A habilidade de entender o comportamento de uma rede *WirelessHART* é de fundamental importância. Falhas nos dispositivos de transmissão, bloqueios físicos ou interferências devem ser bem estudados para que não comprometam a comunicação da planta.

O objetivo deste trabalho é o desenvolvimento de um software capaz de analisar logs vindos da ferramenta *Wi-Analys*. Nesta nova ferramenta o usuário poderá obter dados importantes por meio de uma interface gráfica intuitiva.

5.1. Interface

Para a construção da interface gráfica será utilizado subsistema gráfico *Windows Presentation Foundation* (WPF) presente no *framework* .NET 3.0.

A biblioteca *GraphSharp* será utilizada para renderizar uma estrutura de grafo. O grafo representará a estrutura da rede e mostrará quais nodos possuem links de comunicação entre si. Já a qualidade dos links será apresentada utilizando arestas coloridas.

Campos de texto mostrarão informações pertinentes a cada nó, são elas:

- RSL entre vizinhos.
- RSL médio entre dispositivo e *Wi-Analys*.
- Número de pacotes recebidos.
- Número de pacotes enviados.
- Número de pacotes perdidos.
- Número de links.

As informações serão agrupadas em períodos de tempo, como se fossem fotografias do estado da rede em um dado momento. Assim o usuário poderá navegar, avançando e retornando, pelos diversos estados assumidos pela rede.

5.2. Entrada e processamento dos Dados de Log

Os logs vindos do analisador *Wi-Analys* serão carregados utilizando a biblioteca *System.IO* do *framework .Net*. Esta biblioteca foi escolhida por oferecer funções de manipulação de fluxos de textos de forma simples e expressiva. A característica estruturada do Log *Wi-Analys* permite que os dados sejam extraídos utilizando um parser simples para gerar a estrutura que guardará as informações. Expressões regulares e funções de manipulação de strings serão utilizadas para capturar os dados vindos dos Logs.

A seguir um diagrama de classes simplificado das entidades do software.

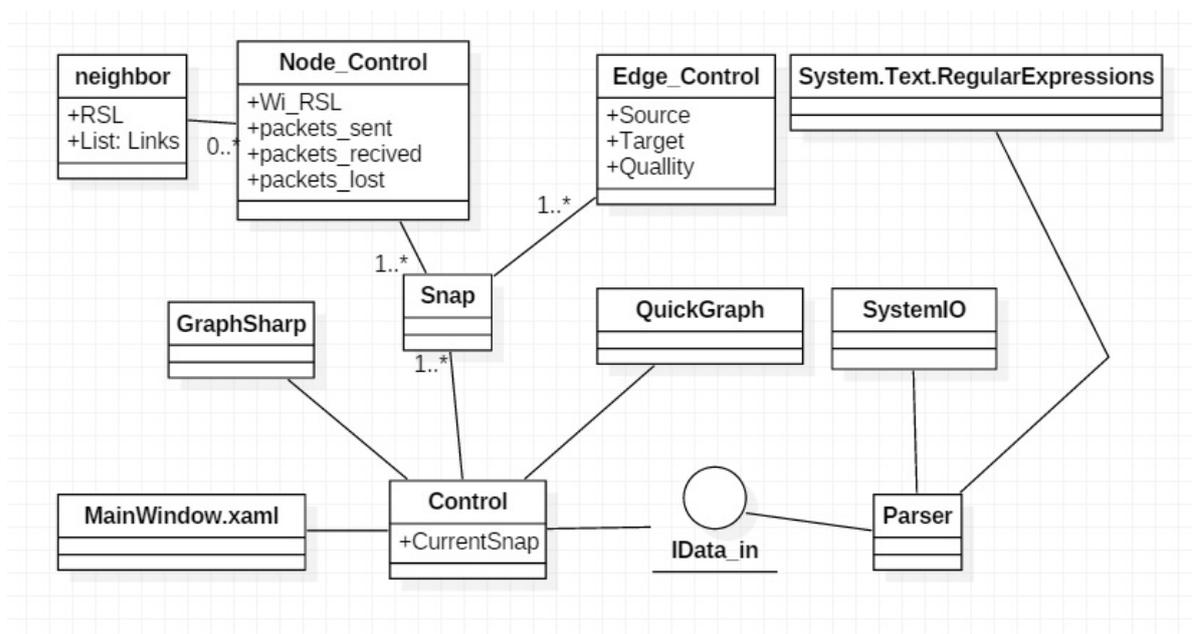


Figura 5. Diagrama de classes simplificado

5.3. Testes

Os testes, para verificar o correto funcionamento da ferramenta, serão realizados com dados de uma rede montada em laboratório. O ambiente controlado permite criar situações, tais como falhas nos dispositivos ou obstruções de sinal, para serem analisadas com o auxílio da ferramenta.

A figura 6 mostra o fluxo do processo de análise de uma rede WH. No passo um os comandos pertinentes são enviados para o *gateway* WH para aquisição de dados relativos aos nós conectados à rede *WirelessHART*, o passo dois recebe essas informações e as guarda para uso da ferramenta de análise. Na sequência, passo três, os logs da ferramenta de captura de rede são recolhidos. Por fim, no passo 4 as informações são inseridas na ferramenta para interpretação do usuário.

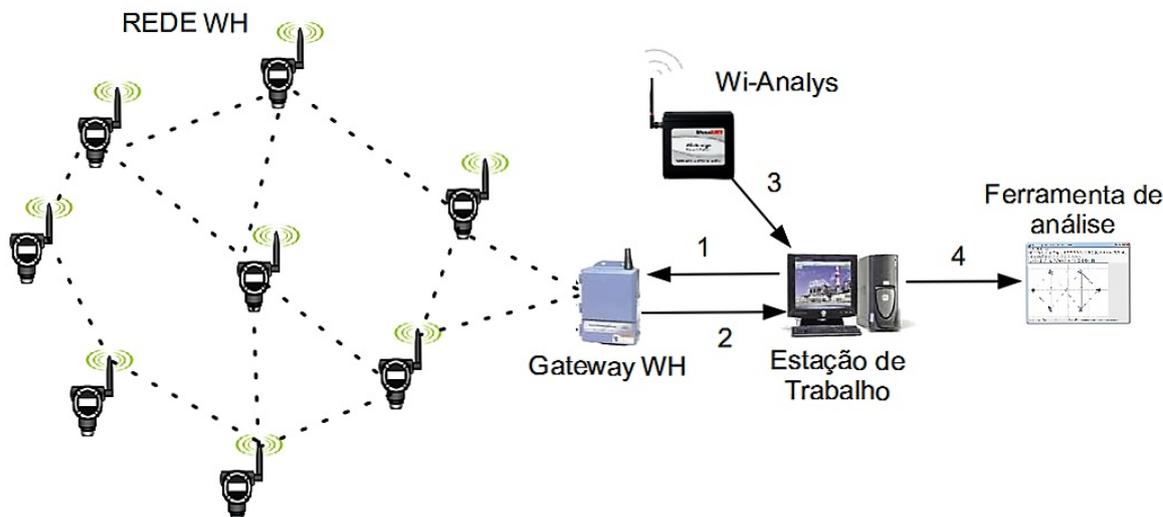


Figura 6. Fluxo de execução da análise de uma rede WH

6. Conclusões e Planejamento

Em sistemas que envolvem riscos a vida ou estruturas com alto valor, a fase de testes deve ser cuidadosamente executada. Como mostrado ao longo deste artigo, o protocolo apresentado é complexo e suas ferramentas para testes precisam ser expandidas para fornecer maior usabilidade aos usuários. O resultado deste trabalho ajudará na análise dos testes, principalmente em experimentos longos que geram uma grande quantidade de dados.

Com o objetivo definido e introduzida a base do protocolo *WirelessHART* podemos seguir para a próxima etapa : A tabela 2 mostra as atividades previstas organizadas ao longo do período de elaboração do Trabalho de Graduação II.

Tabela 2. Tabela com o plano de atividades para o Trabalho de Graduação 2.

Atividade	Abril	Maior	Junho	Julho
Estudos sobre bibliotecas auxiliares	X			
Implementação do parser	X			
Implementação da classe controle		X		
Implementação da classe Construção de interface gráfica		X		
Integração com software para envios dos comandos		X		
Preparação do setup para testes			X	
Execução dos Testes			X	
Monografia		X	X	
Apresentação				X

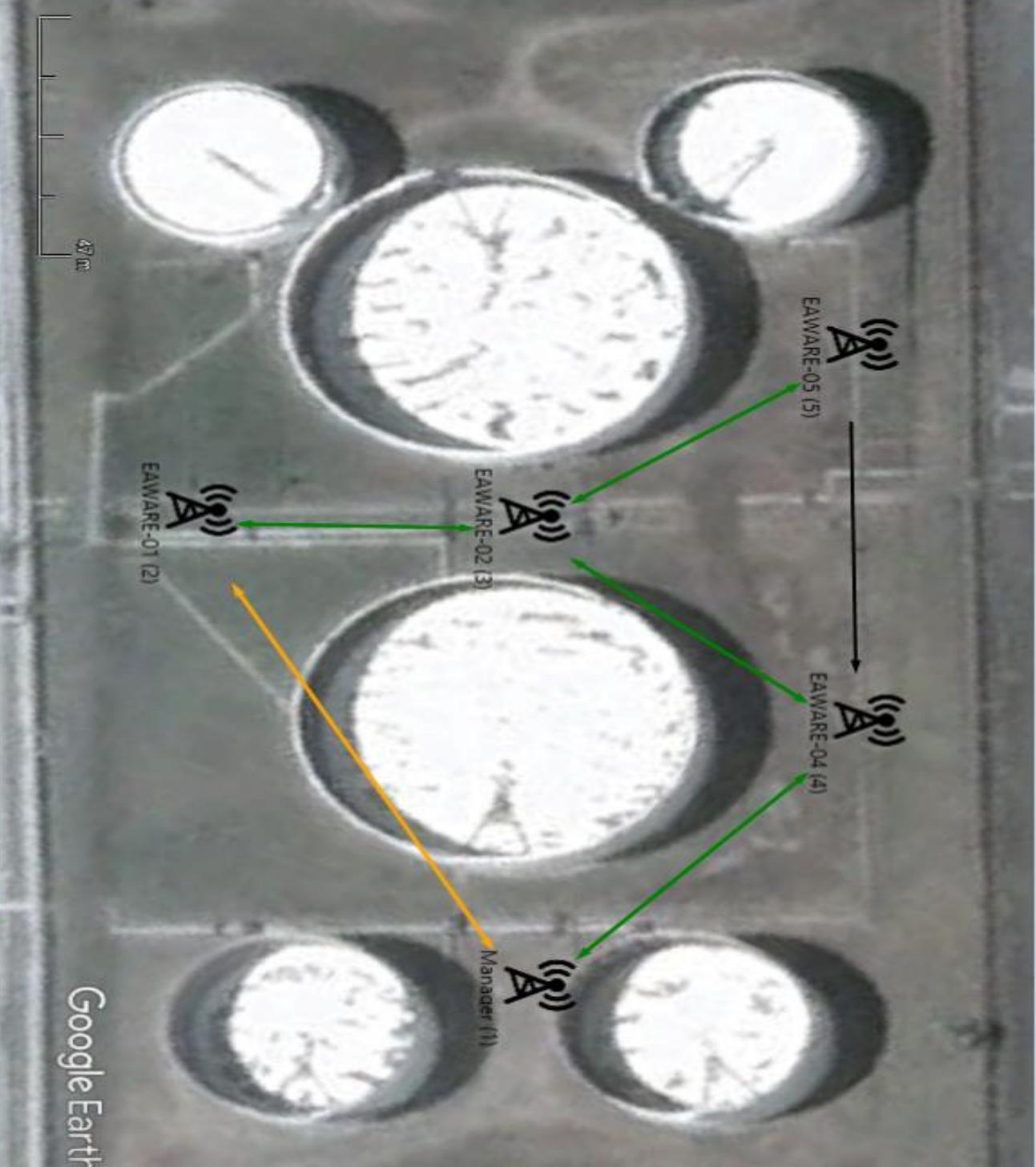
Referências

- Chen, D., Nixon, M., e Mok, A. (2010). *Why WirelessHART*. Springer.
- Kunzel, G. (2012). Ambiente para avaliação de estratégias de roteamento para redes wirelesshart.

- Lorençato, A. d. A. (2013). Analisador de redes wirelesshart.
- Machado, T. M. (2014). Analisador de redes wirelesshart com capacidade de detecção de coexistência.
- SPEC-065 (2007). *2.4GHz DSSS O-QPSK Physical Layer Specification*. HART Communication Foundation.
- SPEC-075 (2008). *TDMA Data Link Layer Specification*. HART Communication Foundation.
- SPEC-085 (2009). *Network Management Specification*. HART Communication Foundation.
- SPEC-127 (2008). *Universal Command Specification*. HART Communication Foundation.
- SPEC-151 (2008). *Common Practice Command Specification*. HART Communication Foundation.
- SPEC-155 (2008). *Wireless Command Specification*. HART Communication Foundation.
- Winter, J. M. (2010). Software de análise de roteamento de dispositivos wirelesshart.

ANEXO B — VISUALIZADOR DE LOGS WI-ANALYS

- Título: Interface Gráfica-Visualizador de *logs Wi-Analys*.
- Data: Novembro de 2016



Name: EAWARE-01

Nick: 2

Info Links

To	SF_ID	Slot	offset	Chan	Type	Xmit	Recv	Share
65535	0	1	0		Discovery	True	True	False
65535	1	31	7		Join	True	False	False
65535	4	38	13		Broadcast	True	False	False
65535	0	264	9		Join	False	True	False
1	1	50	5		Broadcast	False	True	False
1	0	238	12		Normal	True	False	False
1	0	748	12		Normal	True	False	False
1	0	1000	12		Normal	True	False	False
3	0	288	0		Normal	False	True	False
3	0	551	11		Normal	False	True	False

Linked Neighbors

Neighbor	Transmissions	Recv	Failures	MeanRSL
Manager	479	203	139	-72
EAWARE-02	44	659	0	-66

Unlinked Neighbors

EAWARE-04---> Mean RSL: -7

Total Communication Values

Transmissions: 523
 Received: 862
 Failures: 139

Current Snap: 2
 Snap Time (min):
 Test Time (min): 60

Unknown
 Failure < 25%
 25% > Failure < 50%
 Failure > 50%

Google Earth

ANEXO C — GUIA DE UTILIZAÇÃO DO SOFTWARE

- Título: Guia de Utilização do *Software - Wi-Analys Log Visualizer*.
- Data: Dezembro de 2016

Universidade Federal do Rio Grande do Sul

Guia de Utilização do Software
Wi-Analys Log Visualizer

Matheus Maia de Souza

Porto Alegre

2016

1. Introdução

O software *Wi-Analys Log Visualizer* oferece condições que facilitam a análise de logs obtidos com o *sniffer* para redes *WirelessHart*, *Wi-Analys*.

Com o *Wi-Analys Log Visualizer* é possível:

- Visualizar a topologia de uma rede *WirelessHart*;
- Visualizar os links de um determinado dispositivo.
- Obter os valores de transmissões, falhas de transmissões e recepções totais de qualquer dispositivo de uma rede *WirelessHart*;
- Obter os valores de transmissões, falhas de transmissões, recepções e MeanRSL de um certo dispositivo com seus vizinhos linkados.
- Obter os valores, percebidos por um certo dispositivo, de MeanRSL dos vizinhos não linkados.

Para realizar a visualização é necessário que o usuário possua um arquivo de *log* gerado pelo *Wi-Analys*, com seus dados devidamente decriptados.

2. Utilização

A figura a seguir mostra a interface do software, na qual são recebidos os comandos do usuário e apresentados os dados pertinentes.

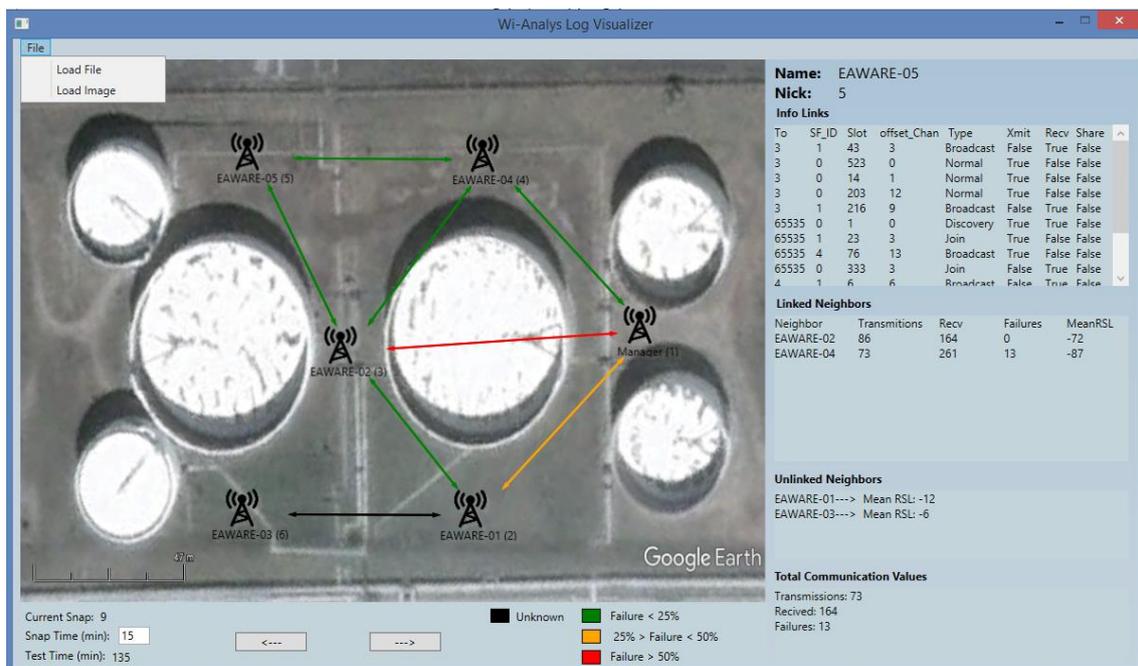


Figura 1 Interface de usuário *Wi-Analys Log Visualizer*

Para dar início a visualização o usuário deve seguir os seguintes passos:

1. Definir o período de tempo coberto por um Snap (Snap Time). Para isso o usuário deve inserir o valor, em minutos, desejado na caixa de texto localizada na parte inferior esquerda da interface de usuário.
2. Localizar o arquivo *log*. Essa operação é realizada acessando o menu File e em seguida o menu Load File.
3. Caso o usuário deseje inserir uma figura de fundo para a topologia da rede, deve acessar o menu File e em seguida acessar o menu Load Image.
4. Posicionar os dispositivos. Este passo deve ser realizado toda vez que surgir um novo dispositivo na rede. O processo de posicionamento é realizado selecionando o dispositivo e arrastando-o até a posição desejada. Uma vez posicionado, o dispositivo permanece em seu lugar até que seja arrastado para uma nova posição.
5. Avançar/Retornar Snap: para avançar para o próximo Snap ou retornar a um Snap já visualizado, o usuário deve acessar os botões localizados na parte inferior da interface de usuário. Os botões seta para esquerda e seta para direita são, respectivamente, para retornar e avançar um Snap.
6. Obter informações sobre dispositivo. Após terminada a leitura do arquivo log, o usuário pode acessar as informações individuais dos dispositivos. Para isso, basta clicar no dispositivo correspondente apresentado na topologia, apresentada no centro da interface de usuário. As informações são mostradas em painéis a esquerda de topologia. O dispositivo acessado é identificado nos campos Name e Nick, em seguida, são exibidos os links pertencentes ao dispositivo, informações dos vizinhos linkados, informações dos vizinhos não linkados e informações totais sobre as transmissões, falhas e recepções do dispositivo, em questão.

As cores das arestas entre os dispositivos representam a proporção de falhas no total de transmissões entre vizinhos. Os valores arbitrados e seus conceitos de qualidade são mostrados na tabela a seguir:

Qualidade	Proporção	Cor
Boa	Menor que 25 %	Verde
Intermediária	Maior que 25 %	Laranja
Intermediária	Menor que 50%	Laranja
Ruim	Maior que 50%	Vermelho

Nota: Para arestas bidirecionais o usuário deve ter atenção para os valores individuais de cada sentido, uma vez que é realizada uma média das falhas e transmissões totais entre os dois dispositivos. Esses valores individuais são obtidos no painel Linked Neighbors.