UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA PROGRAMA DE
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

TATIANE MARTINS MACHADO

ANALISADOR DE REDES *WIRELESS*HART COM CAPACIDADE DE DETECÇÃO DE COEXISTÊNCIA

Porto Alegre
2014

TATIANE MARTINS MACHADO

ANALISADOR DE REDES *WIRELESS*HART COM CAPACIDADE DE DETECÇÃO DE COEXISTÊNCIA

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica.

Área de concentração: Controle e Automação

ORIENTADOR: Prof. Dr. Carlos Eduardo Pereira

Porto Alegre

2014

TATIANE MARTINS MACHADO

ANALISADOR DE REDES *WIRELESS*HART COM CAPACIDADE DE DETECÇÃO DE COEXISTÊNCIA

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia Elétrica e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____

Prof. Dr. Carlos Eduardo Pereira, UFRGS

Doutor pela Universität Stuttgart – Stuttgart, Alemanha

Banca Examinadora:

Prof. Dr. Walter Fetter Lages, UFRGS

Doutor pelo Instituto Tecnológico de Aeronáutica – São Paulo, Brasil

Prof. Dr. Valner João Brusamarello, UFRGS

Doutor pela Universidade Federal de Santa Catarina – Florianópolis, Brasil

Prof. Dr. Fabiano Passuelo Hessel, PUCRS

Doutor pela Université Joseph Fourier – Grenoble, França

Coordenador do PPGEE: _____

Prof. Dr. Arturo Suman Bretas

Porto Alegre, março de 2014.

AGRADECIMENTOS

Ao professor Carlos Eduardo Pereira, pela oportunidade de realização deste trabalho.

Aos colegas Ivan Müller, Jean Michel Winter e Alexandre Lorençato pelo auxílio no desenvolvimento deste trabalho e ajuda na realização dos testes práticos.

Aos amigos e familiares que me apoiaram e me incentivaram durante este período.

Ao meu namorado Victor Hugo Dickow, pelo incentivo, paciência e apoio para a conclusão desta etapa.

RESUMO

O uso de redes sem fio no ambiente industrial está crescendo cada vez mais devido às vantagens que estas redes apresentam comparadas aos sistemas cabeados. No entanto, é necessário que as redes sem fio apresentem a mesma robustez que as redes cabeadas. Para isso, é necessário garantir o enlace de rádio frequência entre os dispositivos e também é preciso que a rede atenda aos requisitos de comunicação em tempo real. O protocolo *Wireless*HART foi criado em 2007, sendo o primeiro padrão aberto de comunicação sem fio especialmente desenvolvido para ambientes industriais. Esse protocolo vem ganhando cada vez mais aplicabilidade na indústria devido à sua alta confiabilidade e robustez. Por se tratar de um protocolo recente, ainda existem muitas pesquisas em andamento, sendo uma das áreas relacionada a ferramentas para análise e monitoramento da rede. Tais ferramentas têm como objetivo principal capturar as mensagens que trafegam na rede, ou seja, atuam como um *sniffer*. No entanto, o protocolo *Wireless*HART apresenta algumas peculiaridades quando comparado a outras redes sem fio, como por exemplo, o uso de 15 canais diferentes. Devido a isso, muitos *sniffers* já propostos para redes sem fio não podem ser utilizados, pois capturam dados em apenas uma frequência. Outra questão está relacionada ao monitoramento de redes já instaladas, onde os dispositivos podem se encontrar em locais de difícil acesso. Dessa forma, é necessário que a ferramenta tenha mobilidade, ou seja, não necessite estar conectada a um computador, por exemplo. Então, uma vez que ainda não existe uma ferramenta de monitoramento ideal para redes *Wireless*HART, este trabalho propõe um nova ferramenta, que apresenta vantagens, tais como: captura de dados nos 15 canais utilizando apenas um receptor de rádio, armazenamento local dos dados capturados utilizando um cartão de memória e medição da energia nos canais, para obter informações a respeito de interferências na rede. Além disso, uma aplicação *offline* de análise dos dados apresenta estatísticas e análises a respeito dos dados capturados. Este trabalho apresenta alguns conceitos teóricos importantes a respeito do protocolo e os detalhes da implementação da ferramenta. Os resultados obtidos mostram que a ferramenta está atuando corretamente como um *sniffer* e está realizando a detecção de interferências na rede. Um estudo de caso mostra o comportamento da rede *Wireless*HART coexistindo com o padrão IEEE 802.11.

Palavras-chaves: *Wireless*HART, analisador de redes, coexistência em redes sem fio.

ABSTRACT

The use of wireless networks in industrial environment is growing due to the advantages of these networks compared to wired systems. However, the wireless networks must have the same robustness that wired networks. It is necessary to ensure the link between devices and it is also necessary that the network meets the requirements of real-time communication. The *Wireless*HART protocol was created in 2007, and it is the first open wireless communication standard specifically designed for industrial environments. This protocol has been gaining increasing applicability in industry due to its high reliability and robustness. Because it is a recent protocol, there are still many ongoing researches. One of them is related to tools for analyzing and monitoring the network. Such tools have the main objective to capture the messages that travel on the network, acting as a sniffer. However, the *Wireless*HART protocol has some peculiarities as compared to other wireless networks, such as the use of 15 different channels. Because of this, many sniffers already proposed for wireless networks can not be applied, because they capture data on only one frequency. Another issue is related to the monitoring networks already installed, where devices can be found in places of difficult access. Thus, it is necessary that the tool has mobility, i.e., does not need to be connected to a computer, for example. So, since there is still no ideal monitoring tool for *Wireless*HART networks, this paper proposes a new tool, which has advantages such as: data capture in 15 channels using only one radio receiver, local storage of captured data using a memory card and measuring the energy in the channels for getting information about interference in the network. In addition, an offline application presents statistics and analysis about the captured data. This paper presents some important theoretical concepts about the protocol and implementation details of the tool. The results show that the tool is working properly as a sniffer and it's performing the interference detection in the network . A case study shows the behavior of the *Wireless*HART network coexisting with IEEE 802.11 standard.

Keywords: *Wireless*HART, network analyzer, coexistence in wireless networks.

SUMÁRIO

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS

ACK: *Acknowledge*

ASN: *Absolute Slot Number*

CCA: *Clear Channel Assessment*

CH: *Channel*

CRC: *Cyclic Redundancy Check*

CS: *Chip Select*

DLPDU: *Data-Link Protocol Data Unit*

ED: *Energy Detection*

HCF: *HART Communication Foundation*

IP: *Internet Protocol*

LQI: *Link Quality Indicator*

OSI: *Open Systems Interconnection*

PHY: Interface física

RF: Rádio frequência

RSL: *Received Signal Level*

SD: *Secure Digital*

SPI: *Serial Peripheral Interface*

TDMA: *Time Division Multiple Access*

UART: *Universal Asynchronous Receiver/Transmitter*

UDP: *User Datagram Protocol*

WH: *Wireless*HART

1 INTRODUÇÃO

O controle e o monitoramento de processos no ambiente industrial vêm sofrendo mudanças nos últimos anos, devido ao crescimento e avanço das pesquisas relacionadas a redes sem fio para indústria. As redes sem fio apresentam muitas vantagens sobre as redes cabeadas, entre elas: (i) grande redução ou ausência de custo para instalação dos cabos; (ii) os nós da rede podem ser colocados em qualquer ponto da planta, onde muitas vezes não seria possível a colocação de cabos; (iii) baixo custo de manutenção, uma vez que não existe o custo associado à movimentação e troca de cabos; (iv) a rede pode ser aumentada sob demanda, sem a necessidade de alteração da infraestrutura pré-existente.

Entretanto, o ambiente industrial exige alta confiabilidade, ou seja, é necessário haver uma perfeita comunicação entre os dispositivos da rede, os requisitos temporais devem ser cumpridos e a integridade e segurança das mensagens que trafegam pela rede devem ser garantidas. Todas estas características devem ser atendidas sob as condições críticas do ambiente industrial, tais como: temperaturas elevadas, umidade, ruídos eletromagnéticos e obstáculos físicos (JONSSON; KUNERT, 2009). Estas condições impostas pelo ambiente industrial são os principais fatores que determinam as desvantagens das redes sem fio em relação às redes cabeadas, pois estas últimas apresentam maior robustez nestes ambientes, enquanto as redes sem fio sofrem com problemas de interferências nos sinais.

Apesar das vantagens das redes sem fio já serem conhecidas há anos, foi somente com a disponibilidade de dispositivos de rádio frequência (RF) de baixo custo que a ideia de redes sem fio industriais saiu dos laboratórios de pesquisa, transformando-se em produtos comerciais. Em particular, a criação da norma IEEE 802.15.4 (2003) para redes pessoais de baixa taxa de dados (LR-PAN – *Low Rate Personal Area Network*) rapidamente se tornou uma solução promissora para definir padrões para as redes sem fio industriais, pelo menos como referência para projetos de *harware* (DEPARI et al., 2009).

Muitos protocolos industriais para comunicação sem fio têm sido propostos, como por exemplo: ZigBee PRO, ISA SP 100.11a, WIA-PA e *Wireless*HART (MULLER; PEREIRA; NETTO, 2011), sendo que todos esses protocolos possuem seus transceptores compatíveis com a camada física do padrão IEEE 802.15.4. Por outro lado, as camadas superiores precisam sofrer mudanças para que os requisitos da aplicação sejam satisfeitos.

O protocolo *Wireless*HART (WH) foi criado em 2007 e foi o primeiro padrão aberto para comunicação sem fio projetado especialmente para o controle de processos industriais em tempo real. Por se tratar de um protocolo recente, ainda há muitas pesquisas em andamento, a fim de proporcionar um melhor entendimento e melhorias no protocolo, facilitando, consequentemente, sua aplicabilidade em ambientes reais. Além disso, há muitos pontos a serem explorados, pois a norma do protocolo muitas vezes não define rigorosamente como determinada funcionalidade deve ser implementada, mas sim define regras a serem seguidas, de forma que o desenvolvedor do dispositivo WH tem a oportunidade de realizar melhorias em diversos aspectos do protocolo.

Então, juntamente com esse cenário atual de muitas pesquisas e aperfeiçoamento do protocolo WH, surge a necessidade de ferramentas que permitam aos pesquisadores, desenvolvedores de soluções *Wireless*HART, engenheiros de operação de plantas industriais que utilizem o protocolo e usuários monitorar, avaliar e inspecionar a rede em operação. Uma das ferramentas primordiais são os *sniffers*, que capturam as mensagens que trafegam pela rede.

Um *sniffer* está relacionado diretamente com a camada física de um protocolo. O WH possui muitas peculiaridades referentes à sua camada física, como por exemplo, o uso de 15 canais do padrão IEEE 802.15.4 e saltos entre estes canais a cada nova comunicação. Dessa forma, grande parte dos *sniffers* propostos para redes sem fio em geral não são aplicáveis em redes WH, pois não tem a capacidade de capturar dados nos 15 canais. Outro fato importante

a ser considerado é que as mensagens WH são criptografadas e, dessa forma, somente a captura dos dados por um *sniffer* não é suficiente, uma vez que não será possível verificar o conteúdo da mensagem. Portanto, além da captura das mensagens, é necessário um mecanismo que garanta que as mesmas poderão ser decifradas.

A *HART Communication Foundation* (HCF), que é a criadora do protocolo WH, criou um *sniffer* para desenvolvimento de dispositivos compatíveis, denominado *Wi-Analys*, que realiza a captura dos dados em todos os canais, através da utilização de 16 receptores de rádio, e realiza a decriptação dos dados. É necessário que o *sniffer* esteja conectado a um computador, onde os dados são armazenados e onde a decriptação é, de fato, realizada. Nenhuma análise a respeito dos dados é realizada. Essa ferramenta é muito útil para uso em laboratórios, mas a necessidade de um computador próximo ao ponto de medição muitas vezes inviabiliza seu uso no monitoramento de redes WH já instaladas em campo.

Além do *sniffer* da HCF, existem apenas propostas, na literatura, de novos *sniffers* para redes WH. Muitos trabalhos tratam apenas da captura das mensagens, não abordando o tratamento dos dados, sendo que muitos não realizam nem mesmo a decriptação das mensagens. Algumas propostas apresentam *hardware* bastante complexo em virtude da necessidade de capturar as mensagens em 15 canais diferentes.

Dessa forma, ainda existe a necessidade de desenvolvimento de uma ferramenta capaz de fornecer ao usuário informações mais detalhadas a respeito da rede e com possibilidade de ser utilizada em redes já instaladas.

Outra questão importante a ser considerada durante a operação de uma rede WH diz respeito à coexistência do WH com outros protocolos. O WH utiliza a faixa de frequência não licenciada de 2,4 GHz (ISM – *Industrial, Scientific, Medical*), que é uma faixa vastamente explorada. Por exemplo, a família IEEE 802.11 (Wi-Fi) também opera nesta frequência. Dessa forma, uma ferramenta que permita analisar as interferências na rede é bastante útil ao

usuário, até mesmo porque o WH permite o uso da lista de canais proibidos (*blacklist*), onde o usuário pode selecionar canais para serem evitados durante a operação da rede. Assim, os canais que sofrem interferências podem ser removidos da lista de canais ativos da rede. Outra possibilidade ainda não explorada é o fato de que a própria rede poderia se adaptar e remover os canais com interferência automaticamente.

Tendo em vista as necessidades que ainda existem na área de monitoramento e análise de redes WH, este trabalho propõe o desenvolvimento de uma ferramenta que permite capturar as mensagens que estão trafegando na rede, ao mesmo tempo em que a interferência na rede é medida. A proposta apresenta a vantagem de não necessitar de 15 rádios receptores. Em vez disso, utiliza-se apenas um rádio que é programado para se adaptar à dinâmica da rede e capturar os dados nos 15 canais. Dessa forma, tem-se um *hardware* bastante simples.

Para que a ferramenta proposta possa ser aplicada em redes já existentes, as quais muitas vezes possuem dispositivos em locais de difícil acesso, os dados capturados são armazenados em um cartão de memória para posterior análise. O armazenamento dos dados é feito por um segundo rádio, que também é utilizado para realizar a medição de energia nos canais da rede, de forma a detectar interferências. A medição de energia nos canais é o fator que determina a necessidade de dois rádios, uma vez que a energia deve ser medida ao mesmo tempo em que uma mensagem estiver sendo recebida, fazendo com que sejam necessários dois transceptores sintonizados no mesmo canal para realizar estas tarefas.

Além da captura das mensagens e detecção de interferências, também são feitas análises dos dados, as quais são apresentadas ao usuário através de uma ferramenta de análise *offline*. Este também é um diferencial da ferramenta proposta neste trabalho em relação às ferramentas propostas na literatura, as quais não realizam o tratamento e análise dos dados.

1.1 OBJETIVOS

Os objetivos deste trabalho são os seguintes:

a) desenvolver uma ferramenta de monitoramento e análise de redes WH, capaz de capturar as mensagens que trafegam pela rede (que estejam ao alcance do ponto de coleta dos dados) e identificar possíveis interferências na rede;
b) gerar análises e estatísticas a respeito dos dados capturados, de forma a apresentar ao usuário, de maneira clara, as informações que podem ser extraídas dos mesmos.

Esta dissertação é dividida da seguinte forma: o capítulo 2 apresenta uma conceituação teórica sobre alguns aspectos do protocolo WH que são importantes para o entendimento da ferramenta proposta; o capítulo 3 apresenta uma análise do estado da arte sobre ferramentas para monitoramento e análise de redes WH; o capítulo 4 apresenta em detalhes a ferramenta proposta, sendo tratada tanto a captura quanto a análise dos dados; o capítulo 5 apresenta as validações experimentais e resultados obtidos, e o capítulo 6 apresenta as conclusões e propostas de trabalhos futuros.

2 CONCEITUAÇÃO TEÓRICA

Neste capítulo, são apresentados alguns conceitos importantes para o entendimento do trabalho desenvolvido. Não são abordados conceitos fundamentais do protocolo WH, tais como o detalhamento de cada uma das suas camadas (baseado no modelo OSI), uma vez que já existem muitos trabalhos na literatura que trazem esta fundamentação (KUNZEL, 2012; CHEN, 2010). Adicionalmente, outras dissertações e teses elaboradas por alunos que desenvolveram suas pesquisas no Grupo de Controle, Automação e Robótica (GCAR-UFRGS) (MULLER, 2012; WINTER, 2013) também apresentam esta fundamentação. O foco deste capítulo é a camada física e camada de enlace da rede WH, os tipos de mensagens WH e as informações que podem ser extraídas destas, bem como o processo de agregação (*join*) de dispositivos à rede.

2.1 CAMADA FÍSICA

A camada física do WH é baseada na norma IEEE 802.15.4 (IEEE, 2011), operando na faixa de frequência livre de licença ISM (*Industrial*, *Scientific* e *Medical*) de 2,4 GHz, a uma taxa de 250 kbits/s. Utiliza a técnica de *Channel Hopping* (salto de canais), fazendo com que cada comunicação entre dois dispositivos ocorra em uma frequência distinta. Essa técnica é empregada para reduzir a possibilidade de interferências com outras redes sem fio.

A norma IEEE 802.15.4 define os canais 11 a 26 para o espectro de 2,4 GHz, sendo que cada canal possui uma largura de banda de 2 MHz, espaçados a cada 5 MHz. Porém, como o canal 26 é empregado em outros sistemas de RF em alguns locais do planeta, o WH utiliza apenas os canais 11 a 25. A Tabela 1 apresenta as frequências centrais dos canais utilizados no WH.

Tabela 1 – Frequências centrais dos canais do WH

Canal	Freq. (GHz)	Canal	Freq. (GHz)	Canal	Freq. (GHz)
11	2,405	16	2,430	21	2,455
12	2,410	17	2,435	22	2,460
13	2,415	18	2,440	23	2,465
14	2,420	19	2,445	24	2,470
15	2,425	20	2,450	25	2,475

Fonte: HART COMMUNICATION FOUNDATION, 2007

Todos os dispositivos WH devem fornecer uma EIRP (potência radiada isotrópica equivalente) nominal de até +10 dBm e devem possuir antena omni-direcional. O nível de potência do dispositivo deve ser controlado (programável) em níveis discretos de -10 dBm a +10 dBm EIRP.

As distâncias de comunicação esperadas para ambientes *outdoor* (com linha de visada direta) e ambientes *indoor* (sem linha de visada direta) são mostradas na Tabela 2.

Tabela 2 – Distâncias de comunicação do WH

Ambiente *outdoor*		Ambiente *indoor*	
Potência de transmissão (dBm)	Distância (m)	Potência de transmissão (dBm)	Distância (m)
0	100	0	35
10	200	10	75

Fonte: HART COMMUNICATION FOUNDATION, 2007

A camada física possui as seguintes funcionalidades (IEEE, 2011):

Detecção de energia (ED – *Energy Detection*): é uma estimativa da potência do sinal recebido dentro da largura de banda do canal. Nenhuma tentativa é feita para identificar ou decodificar sinais no canal. O tempo de medição do ED deve ser igual ao período de 128 us.

Avaliação da utilização do canal (CCA – *Clear Channel Assessment*): avalia se determinado canal está em uso. A norma IEEE 802.15.4 define 6 tipos de CCA, sendo que o CCA utilizado no WH é do tipo 2 (*Carrier Sense Only* – sensibilidade da portadora somente). Neste caso, o CCA deve reportar o meio ocupado somente quando for detectado um sinal compatível com a norma IEEE 802.15.4, com a mesma modulação e características de espalhamento do PHY que está em uso no dispositivo.

Indicador de qualidade do *link* (LQI – *Link Quality Indicator*): é uma caracterização da intensidade e/ou qualidade do pacote recebido. A medida pode ser feita utilizando o ED recebido, uma estimativa da relação sinal-ruído, ou uma combinação dos dois métodos.

2.2 CAMADA DE ENLACE

O WH utiliza TDMA (*Time Division Multiple Access* – Acesso Múltiplo com Divisão no Tempo) para controle de acesso ao meio, de forma a se obter comunicações determinísticas e livres de colisões. O TDMA é baseado em faixas de tempo (*time slots*), onde ocorrem as comunicações entre os dispositivos. Uma série de *time slots* forma um *superframe*. O tamanho dos *time slots* e o comprimento do *superframe* (em número de *time slots*) são fixos e determinam o ciclo da rede com uma taxa fixa de repetição. No WH, o *time slot* tem duração de 10 ms (HCF, 2008).

Tipicamente, dois dispositivos são associados a um dado *time slot*: um é a origem e o outro é o destino de uma mensagem. No WH, uma transação dentro de um *time slot* inclui a transmissão de um DLPDU (*Data Link Protocol Data Unit* – Unidade de Dados da Camada de Enlace do Protocolo) a partir de um dispositivo de origem, seguido imediatamente pela transmissão de um DLPDU do tipo ACK (*Acknowledgment* - Confirmação) pelo dispositivo de destino. Apenas mensagens que possuam um endereço de destino do tipo *broadcast* (múltiplos receptores) não requerem ACK (HCF, 2008).

A Figura 1 ilustra um *superframe* TDMA e mostra as comunicações que ocorrem em um *time slot*, ou seja, o envio da mensagem pelo dispositivo de origem e a recepção do ACK enviado pelo dispositivo receptor.

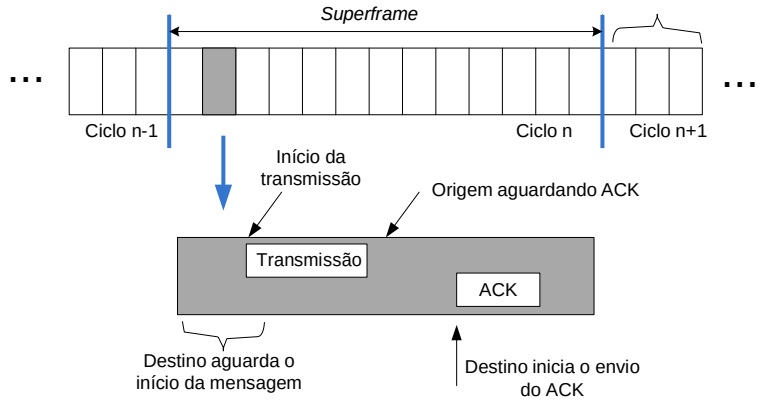


Figura 1 – *Superframe* TDMA e detalhe do *time slot*

Conforme já descrito na seção da camada física, o WH utiliza salto de canais para realizar as comunicações e reduzir as possibilidades de interferências na rede. O salto de canais é realizado a cada *time slot*, como pode ser visto na Figura 2.

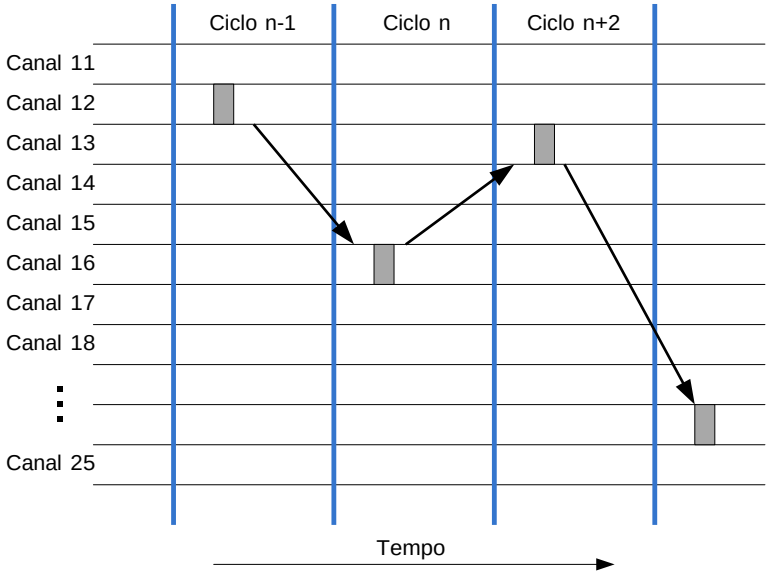


Figura 2 – Salto de canais

O WH possui ainda a lista de canais proibidos, que permite ao administrador da rede selecionar alguns canais para que não sejam usados durante o funcionamento da rede. Isso é útil quando se sabe quais faixas do espectro de frequência estão em uso e, assim, evita-se a utilização dos canais que se sobrepõem à faixa ocupada.

Como já mencionado, uma transação em um *time slot* inclui a transmissão de um DLPDU a partir da origem e a recepção do ACK. A Figura 3 mostra as temporizações que envolvem estas comunicações do ponto de vista do dispositivo de origem e destino e, logo após, as temporizações são explicadas (HCF, 2008).

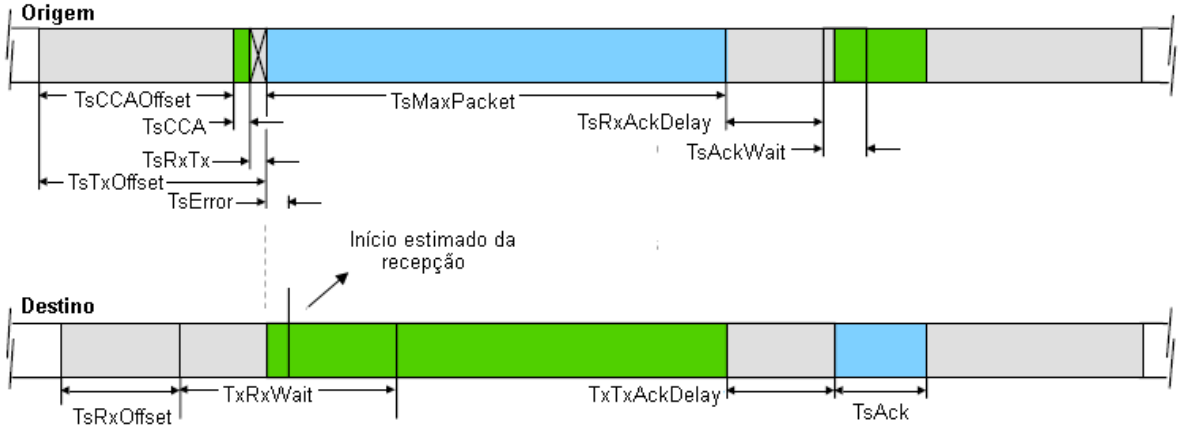


Figura 3 – Temporização de um *time slot*

O *time slot* do dispositivo de origem inicia com um tempo reservado para preparar o pacote a ser transmitido, ou seja, formatação do pacote, cálculo do CRC, entre outros. Isso deve ocorrer durante o tempo *TsCCAOffset*. Após isso, realiza o CCA durante o tempo *TsCCA*, alterna para o modo de transmissão durante o tempo *TsRxTx* e transmite o pacote. Se o CCA detectar que o canal está ocupado, a tentativa de comunicação é reescalonada para um *time slot* posterior. O dispositivo de origem deve iniciar sua transmissão de tal forma que o início da mensagem ocorra exatamente *TsTxOffset* após o início do *time slot*.

Já o dispositivo de destino deve estar no modo de recepção e deve começar a “escutar” por comunicação a partir de *TsRxOffset* do início do *time slot*. A escuta por comunicação ocorre antes e depois do ponto inicial ideal estimado pelo dispositivo receptor. Esta janela de recepção (*TsRxWait*) permite uma flutuação (*drift*) na temporização do dispositivo, fazendo com que ainda assim os dispositivos se comuniquem e resincronizem seus *time slots*. A flutuação na temporização pode ser decorrente da temperatura, envelhecimento do dispositivo, entre outros fatores.

Quando o receptor detecta uma mensagem, o tempo de início da mensagem é capturado (fim da recepção do delimitador da camada física) e a diferença entre o tempo ideal e real do início da transmissão do pacote (*TsError*) é calculado. Caso um dos dispositivos seja fonte de sincronização para o outro, então o tempo final do *time slot* será alinhado após a comunicação ter ocorrido com sucesso.

O dispositivo destino recebe a mensagem e verifica se está endereçada a ele. Caso esteja, o dispositivo deve comutar para o modo de transmissão e iniciar o envio do ACK no tempo *TsTxAckDelay* após o fim da recepção da mensagem enviada pela origem. Enquanto isso, o dispositivo origem está comutando seu transceptor para o modo de recepção e deve "escutar" por comunicação até *TsRxAckDelay* após o fim da sua transmissão. Irá esperar pelo início do ACK por uma duração de *TsAckWait*.

O pacote é considerado enviado com sucesso somente quando a mensagem e o ACK foram recebidos com sucesso pelos dispositivos destino e origem, respectivamente. Se a comunicação for *broadcast*, não existe o envio do ACK.

A Tabela 1 apresenta os tempos especificados pela norma.

Tabela 3 – Temporização do *time slot*

Temporização	Descrição	Valor
TsTxOffset	Tempo entre o início do *time slot* e o início da transmissão da mensagem	2120 μs ± 100 μs
TsRxOffset	Tempo entre o início do *time slot* e o início da escuta por comunicação	1120 μs ± 100 μs
TsRxWait	Tempo de espera pelo início da mensagem	2200 μs ± 100 μs
TsMaxPacket	Comprimento máximo do pacote (inclui cabeçalho,ou seja, 133 bytes)	4256 μs
TsTxAckDelay	Tempo entre fim da recepção e início da transmissão do ACK	1000 μs ± 100 μs
TsRxAckDelay	Tempo entre o fim da transmissão e início da escuta pelo ACK	900 μs ± 100 μs
TsAckWait	Mínimo tempo de espera do início do ACK	400 μs ± 100 μs
TsAck	ACK (26 bytes)	832 μs
TsCCAOffset	Tempo entre o início do *time slot* e o início do CCA	1800 μs ± 100 μs
TsCCA	Tempo de execução do CCA	128 μs
TsRxTx	Tempo de comutação entre transmissão e recepção	192 μs

2.3 TIPOS DE MENSAGENS WH

Existem 5 tipos de mensagens no protocolo WH (HCF, 2008):

Dados: contém dados dos dispositivos e da rede. A origem e destino para DLPDUs do tipo “dados” é a camada de rede.

Keep-Alive: facilita a manutenção da conexão entre dispositivos vizinhos. O *payload* destas mensagens é vazio. Podem ser usadas para realizar a sincronização da rede (uma vez que o ajuste de tempo será retornado pelo ACK correspondente) e também para confirmar a conectividade entre dispositivos e na descoberta de novos vizinhos.

Advertisement: é usado para convidar novos dispositivos a se unir à rede. Quando um dispositivo deseja se unir à rede, ele espera por estes pacotes e usa as informações contidas no pacote para se sincronizar com a rede e iniciar o processo de agregação à rede (*join*). O pacote de *advertisement* inclui informações básicas sobre a rede: ASN (*Absolute Slot Number*), níveis de segurança suportados pela rede e mapa de canais.

***Disconnect*:** é gerado por dispositivos que estão deixando a rede, ou seja, o dispositivo não estará mais disponível para comunicação e deve ser removido da lista de dispositivos. Todos os links envolvendo este dispositivo devem ser removidos.

ACK (*Acknowledgment*): é transmitido por um dispositivo em resposta à recepção de um DLPDU que não seja *broadcast* e não seja outro ACK, ou seja, os dispositivos receptores devem sempre enviar um ACK em resposta a todos os DLPDUs do tipo *Keep-Alive*, *Data* ou *Disconnect* recebidos com sucesso. O ACK contém um código de resposta que indica se o dispositivo receptor aceitou o DLPDU. Por exemplo, código de resposta igual a zero indica que o dispositivo recebeu a mensagem com sucesso e códigos de resposta diferente de zero podem indicar erros, tais como: *buffers* indisponíveis e prioridade da mensagem muito baixa. Nesses casos de erro, a mensagem recebida é descartada.

Além do código de resposta, o *payload* do ACK contém um campo denominado "Ajuste de Tempo", que é o valor de *TsError* indicado na Figura 3, ou seja, é a diferença entre o tempo esperado para o início da recepção de um pacote e o tempo real em que isso ocorreu. É medido em microsegundos, e pode ser positivo se o DLPDU foi recebido antes do esperado ou negativo, se foi recebido depois do esperado.

Alguns dispositivos selecionados pelo gerenciador da rede são usados como fontes de sincronização de tempo. Quando se recebe um DLPDU de um dispositivo que é fonte de sincronização, o dispositivo que recebeu a mensagem deve ter o seu tempo ajustado, baseando-se na informação recebida. A sincronização de tempo é baseada no tempo de chegada do DLPDU ou no valor do campo "Ajuste de tempo" contido no ACK, dependendo qual dispositivo iniciou a transação.

2.4 PROCESSO DE AGREGAÇÃO DE DISPOSITIVOS À REDE WH

O processo de agregação (*join*) de dispositivos de campo à rede WH inicia a partir do momento em que os elementos centrais da rede: *gateway*, gerenciador de rede e ponto de acesso estão operacionais. Na agregação do primeiro dispositivo de campo, apenas o ponto de acesso realiza a propagação de pacotes de anúncio de rede (*advertisement*). Depois que já existem alguns dispositivos agregados à rede, estes também podem começar a enviar pacotes de anúncio, se o gerenciador autorizá-los (MULLER, 2012).

Antes de qualquer dispositivo ingressar na rede WH, é necessário pré-configurá-lo com os valores da chave de agregação (*join key*) e identificação da rede (*network id*). Isso é feito utilizando a porta de manutenção. Como pode haver mais de uma rede WH, o dispositivo precisa estar configurado com o identificador da rede que deseja se unir. Já a chave de agregação é a primeira chave utilizada pelo dispositivo para encriptar as mensagens

iniciais. O usuário deve configurar o mesmo valor desta chave no gerenciador de rede (CHEN, 2010).

De acordo com (MULLER, 2012), o processo de agregação pode ser dividido em 4 etapas, conforme descrito a seguir:

Anúncio: no pacote de anúncio são enviadas as informações necessárias para um dispositivo se unir à rede: identificação do *superframe* e *links* que o dispositivo poderá utilizar para enviar a requisição de agregação, mapa de canais válidos, ASN atual e o identificador de prioridade de agregação (para o dispositivo anunciante). Este pacote é criptografado com a chave de agregação.

Requisição: após o dispositivo de campo ter recebido um pacote de anúncio, ele responderá com uma requisição de agregação. Nesta resposta, constam os seguintes dados: identificador único do dispositivo (*Unique ID*), a denominação do dispositivo (*long tag*), a intensidade de sinal de RF e o apelido dos dispositivos vizinhos. Com base nestes dados, o gerenciador escolherá por meio de qual vizinho as mensagens de agregação serão trocadas com o novo dispositivo de campo.

Primeira resposta: na primeira resposta, o gerenciador de rede envia três comandos para o novo dispositivo. Os comandos são enviados diretamente ao novo dispositivo, em caso de comunicação direta, ou através de p*roxy*, no caso de comunicação indireta. A autenticação das mensagens ponto a ponto é feita utilizando uma chave pública conhecida (*well-known key*) e a autenticação fim a fim utiliza a chave de agregação. Os comandos enviados são: 961, 962 e 963. O comando 961 define uma nova sessão *unicast* entre o dispositivo e o gerenciador de rede, que será utilizada enquanto a rede existir, a fim de permitir a comunicação encriptada entre estes dois elementos da rede. O comando 962 define um apelido (*nickname*) para o novo dispositivo, definido pelo gerenciador. Esse apelido será utilizado futuramente no lugar do *Unique ID*, na ocasião do envio e respostas de novos comandos. O comando 963 define uma

nova chave de rede em substituição da chave bem conhecida. A nova chave é gerada pelo gerenciador e será utilizada para geração de código de integridade de mensagem (MIC – *Message Integrity Code*) na camada de dados, para verificação da autenticidade das mensagens trocadas ponto a ponto, tornando a rede mais segura.

Segunda resposta: na segunda resposta, o novo dispositivo de campo recebe do gerenciador de rede os comandos de escalonamento da rede, os quais definem as rotas, *superframes* e *links* a serem utilizados. A partir deste ponto, o dispositivo já faz parte da rede e pode ter suas variáveis de processo lidas (ou escritas) pela aplicação através do *gateway*.

3 ANÁLISE DO ESTADO DA ARTE

A análise do estado da arte concentra-se em trabalhos já propostos na literatura referentes ao monitoramento e análise de redes WH.

No trabalho de (LIMA, 2012), foi proposta uma ferramenta para análise local da rede WH através da porta de manutenção, que é um item obrigatório em todos os dispositivos de campo WH. Dessa forma, a porta de manutenção não faz apenas o comissionamento do dispositivo, mas também possibilita uma análise local da rede, através de comunicações do tipo requisição/resposta. É necessário conectar um computador ao dispositivo de campo através da porta de manutenção RS485, e uma aplicação executando neste computador permite que o usuário envie comandos ao dispositivo, tais como: ler lista de vizinhos (comando 780) e ler lista de *superframes* (comando 783). A porta de manutenção é capaz de comunicar-se diretamente com o dispositivo sem sobrecarregar a rede. Desta forma, os dados são obtidos mais rapidamente do que através de consultas ao gerenciador de rede através do *gateway*.

No trabalho de (WINTER, 2011), foi proposta uma ferramenta capaz de obter informações relacionadas ao comportamento da rede WH e aos status dos dispositivos. Isto inclui, por exemplo, a análise dos dispositivos que estão na rede, topologia da rede e *links* entre os dispositivos. Foi desenvolvida uma aplicação, que é executada em um computador, e que utiliza o protocolo HART-sobre-UDP para se comunicar com o gerenciador de rede WH. A aplicação é baseada em comandos HART (assim como o próprio protocolo WH), que são utilizados para obter os dados desejados a respeito da rede. Os comandos são encapsulados no padrão UDP e enviados, através do *gateway*, aos dispositivos de campo, que respondem às requisições.

Todos os dispositivos WH devem ser completamente testados e registrados pela *HART Communication Foundation* (HCF), a fim de garantir a compatibilidade com o protocolo de

comunicação e o cumprimento dos requisitos temporais exigidos pela norma. Para isso, a HCF desenvolveu especificações de testes detalhadas e ferramentas de teste, as quais são explicadas no trabalho de (HAN, 2009). Foram propostas três ferramentas, que juntas fornecem um ambiente de verificação de compatibilidade completo para dispositivos WH:

- *Wi-HTest*: tem como objetivo automatizar a execução dos testes definidos pela HCF para redes WH. É capaz de construir os pacotes de teste em tempo real e enviá-los ao dispositivo em teste através de um rádio IEEE 802.15.4. Os testes podem ser divididos em duas categorias: etapa de união do dispositivo à rede (*join*) e comunicação normal de dados. No processo de *join*, o *Wi-HTest* realiza uma sequência de troca de mensagens com o dispositivo em teste e verifica se o dispositivo se uniu à rede corretamente. Já no caso de comunicação normal de dados, o *Wi-HTest* transmite pacotes de dados corretos ou insere erros nos dados e verifica o comportamento do dispositivo nestas condições.

- *Wi-Analys:* é um *sniffer* para capturar pacotes de redes WH em tempo real. É projetado para capturar todas as transmissões IEEE 802.15.4 operando na faixa de frequência de 2,4 GHz, mas filtrando as transmissões a partir de dispositivos WH. O receptor tem a capacidade de capturar dados nos 16 canais do padrão IEEE 802.15.4 simultaneamente e com uma velocidade de até 1000 mensagens por segundo. O *Wi-Analys* é composto por dezesseis receptores de rádio, um concentrador de dados e um software executando em uma estação de trabalho, o qual armazena todas as mensagens capturadas. As mensagens são mostradas ao usuário através de um formulário *online* (em tempo real) ou *offline* (através de um arquivo de *log* de mensagens anteriormente capturadas).

As mensagens são interpretadas e os campos nas mensagens, da camada física até a camada de aplicação, podem ser mostrados em colunas separadamente. Além disso, as mensagens encriptadas são decifradas e mostradas ao usuário como um texto.

O *Wi-Analys* é um produto *standalone* da HCF para ser usado como uma ferramenta de monitoramento da rede WH em tempo real.

- *Post process suite:* analisa os arquivos de *log* gerados pelo *Wi-Analys* durante a execução dos casos de teste pelo *Wi-HTest* para verificar se o dispositivo seguiu a norma rigorosamente durante o teste, especialmente no que diz respeito aos requisitos temporais.

Das ferramentas propostas pela HCF no trabalho de (HAN, 2009), o *sniffer Wi-Analys* é uma opção para monitoramento de redes WH. No entanto, apresenta a desvantagem de necessitar de uma estação de trabalho conectada ao *sniffer* para a coleta de dados. Isto, em redes reais, pode ser inviável dependendo do local onde se deseja instalar o *sniffer*. Mas a ferramenta apresenta grande utilidade para uso em laboratórios durante o estudo do protocolo WH.

(DEPARI et al., 2009) propôs um analisador de rede WH distribuído. Uma vez que um *sniffer* tem uma área de cobertura limitada, ou seja, pode não abranger todos os dispositivos da rede, foi proposto um instrumento distribuído, baseado no uso de diversos *sniffers* interconectados com uma estação de monitoramento. Os dados coletados pelos diversos *sniffers* são enviados para uma rede de medição e encaminhados à estação de monitoramento, a qual pode trocar informações com o gerenciador de rede e pode executar uma aplicação para visualizar os pacotes capturados. Uma análise detalhada dos dados pode ser feita *offline* por meio de um *software* desenvolvido pelo usuário. No entanto, a rede de medição utilizada pelos autores é uma rede Ethernet, ou seja, cabeada. Este fato se opõe às vantagens oferecidas pelas redes sem fio, ou seja, ao se utilizar redes sem fio, consegue-se instalar sensores em locais de difícil acesso, onde não seria possível a colocação de cabos.

(KRÄTZIG, 2009) propôs um analisador de pacotes para redes sem fio IEEE 802.15.4, independente do protocolo das camadas superiores. Até 16 canais podem ser capturados em paralelo com uma taxa de amostragem de até 3200 amostras por segundo. O analisador é

composto de 3 partes: uma unidade de rádio frequência (RF), uma unidade de processamento do sinal e uma unidade remota de controle. A unidade de RF consiste de 16 transceptores, um para cada canal definido na norma IEEE 802.15.4 na faixa de frequência de 2,4 GHz. Fornece os pacotes recebidos e o valor do nível do sinal recebido (RSSI). A unidade de processamento do sinal adiciona um *timestamp* e armazena os dados em uma memória FIFO. Os dados são enviados, via Ethernet, para a unidade de controle remoto, que é uma aplicação executando em um computador, a qual processa e mostra os dados *online* ou *offline* ao usuário. Esta proposta não trata os dados capturados pelo *sniffer*, ou seja, não faz a decriptação, organização e filtragem dos dados.

(LORENÇATO et al., 2013) propôs um analisador de rede integrado ao dispositivo de campo WH. O sistema proposto realiza a leitura simultânea em todos os canais do espectro de 2,4 GHz definidos na norma IEEE 802.15.4 e armazena as informações em uma memória não volátil (cartão de memória) no próprio dispositivo de campo. O sistema ainda possui um módulo GPS integrado para sincronizar as aquisições e posicioná-las geograficamente quando utilizado em campo aberto. É utilizada uma estrutura de RF com 17 rádios, sendo que 16 são utilizados como rádios de aquisição, cada um sintonizado em um canal específico. Cada rádio é composto por um microcontrolador onde parte da pilha WH responsável pela recepção dos quadros é implementada. Com isso, pode-se fazer uma interpretação local dos dados, descartando bytes de sincronismo, quadros mal formatados, inválidos ou com erros de validação, reduzindo o fluxo das informações capturadas, armazenadas, transferidas e analisadas pela unidade de análise dos dados. O décimo sétimo rádio, que está integrado à unidade de coleta dos dados, tem a habilidade de se unir à rede, rotear pacotes e responder a comandos. Assim, o analisador de rede passa a ser também um dispositivo de campo WH.

Como se pode observar, ainda não existe uma ferramenta ótima e, por isso, novos trabalhos continuam sendo desenvolvidos, dada a importância de se ter uma ferramenta que permita analisar o comportamento dinâmico de uma rede WH em um ambiente real.

Este trabalho propõe uma nova abordagem para a inspeção de redes WH, unindo algumas das características presentes nas ferramentas já apresentadas, tais como: utilização de um cartão de memória para armazenar os dados coletados, evitando a necessidade de um computador próximo ao ponto de coleta de dados, e comunicação com o *gateway* através do protocolo HART-sobre-UDP para obtenção de dados da rede em tempo real. A Tabela 4 apresenta uma classificação das ferramentas apresentadas neste capítulo, tão bem como a ferramenta proposta nesta dissertação.

Tabela 4 – Classificação das ferramentas para análise de redes WH

	Classificação quanto à:			
Ferramenta proposta por:	**Intrusão na rede**	**Aquisição dos dados**	**Mobilidade**	**Tipo de Análise**
LIMA, 2012	Passivo	Requisição/resposta	Conexão com PC	Online/Offline
WINTER, 2011	Ativo	Requisição/resposta	Conexão com PC	Online/Offline
HAN, 2009	Passivo	Captura total	Conexão com PC	Online/Offline
DEPARI et al., 2009	Passivo	Captura total	Conexão com PC	Online/Offline
KRÄTZIG, 2009	Passivo	Captura total	Conexão com PC	Online/Offline
LORENÇATO et al., 2013	Passivo	Captura total	Cartão de memória	Offline
Nesta dissertação	Ativo/passivo	Captura total	Cartão de memória	Online/Offline

A ferramenta proposta é classificada como sendo ativa e passiva, pois a aplicação de gerenciamento gera uma interferência ativa na rede. Esta aplicação é executada somente no início do processo de análise da rede e tem como objetivo obter o mapeamento da rede. Após esta etapa inicial, tem-se uma inspeção passiva, ou seja, os dados são obtidos sem interferir na rede.

Com relação à aquisição dos dados, a captura total é mais eficiente do que a aquisição por meio de requisição/resposta, pois todos os dados que estão trafegando pela rede são capturados, enquanto a requisição/resposta depende de solicitações feitas pelo usuário ou

aplicação de análise. No entanto, a captura total requer um processamento dos dados obtidos de modo a filtrar e selecionar as informações úteis.

Outra vantagem da ferramenta proposta diz respeito à mobilidade, pois não é necessário existir a conexão com um computador no ambiente de coleta dos dados, já que os dados são salvos em um cartão de memória para posterior análise. Este fato é bastante importante, pois o ponto de coleta dos dados pode ser de difícil acesso. Dessa forma, a ferramenta proposta permite a análise *offline* dos dados, pela obtenção dos dados armazenados no cartão de memória, e *online*, pelo envio e resposta de comandos especiais ao dispositivo coletor.

O principal diferencial da ferramenta proposta em relação às demais é a capacidade de medição de energia nos canais em uso, de forma a detectar possíveis interferências na rede.

4 PROPOSTA DE UMA FERRAMENTA DE MONITORAMENTO E ANÁLISE DE REDES *WIRELESS*HART

A ferramenta proposta faz uso do trabalho desenvolvido por (LORENÇATO, 2013), que propôs um analisador de redes WH que tem sua estrutura dividida em duas etapas:

- uma unidade coletora obtém os dados trafegando na rede e armazena-os em memória não-volátil (cartão de memória, por exemplo);
- uma aplicação em um computador realiza a análise dos dados armazenados (*offline*).

Dois tipos de unidades coletoras foram abordadas: uma unidade monocanal, onde tem-se apenas um rádio IEEE 802.15.4 que é programado por uma aplicação de gerência de forma a ser sintonizado nas frequências corretas a cada *time slot*, e uma unidade multicanal, onde são necessários 15 rádios IEEE 802.15.4, sendo cada um sintonizado em um canal WH. Neste caso, não é necessária uma aplicação de gerenciamento, pois haverá rádios disponíveis para captura em todos os canais WH. No trabalho, foi descrito que a utilização da unidade coletora monocanal não permitiu a gravação dos dados coletados no cartão de memória, devido à falta de memória do dispositivo de campo.

A ferramenta proposta neste trabalho utiliza os conceitos da unidade coletora monocanal. Porém, foram feitas modificações de forma a se conseguir obter mais informações da rede. Além disso, também foi criada uma aplicação para análise *offline* dos dados coletados.

Propõe-se utilizar dois rádios IEEE 802.15.4 conectados fisicamente. O primeiro rádio é denominado “rádio coletor” e tem a função de coletar todas as mensagens trafegando na rede que estejam ao seu alcance. Para isso, uma aplicação de gerenciamento precisa programar o rádio coletor. O segundo rádio é denominado “*tracker*” e tem a função de gravar os dados capturados pelo rádio coletor em um cartão de memória e também realizar a

medição de energia nos canais, de forma a identificar interferências na rede. Esta última característica ainda não foi proposta na literatura, e encontra grande aplicabilidade na análise de redes WH.

Os dois rádios encontram-se em *hardwares* diferentes, pois foi utilizado o que já se tinha disponível, ou seja, são *hardwares* desenvolvidos em trabalhos anteriores, conforme será descrito posteriormente. Se fosse feito um novo projeto, poderia se ter apenas um *hardware*.

A Figura 4 ilustra a ferramenta proposta e a seguir serão descritos detalhadamente todos os itens que a compõe.

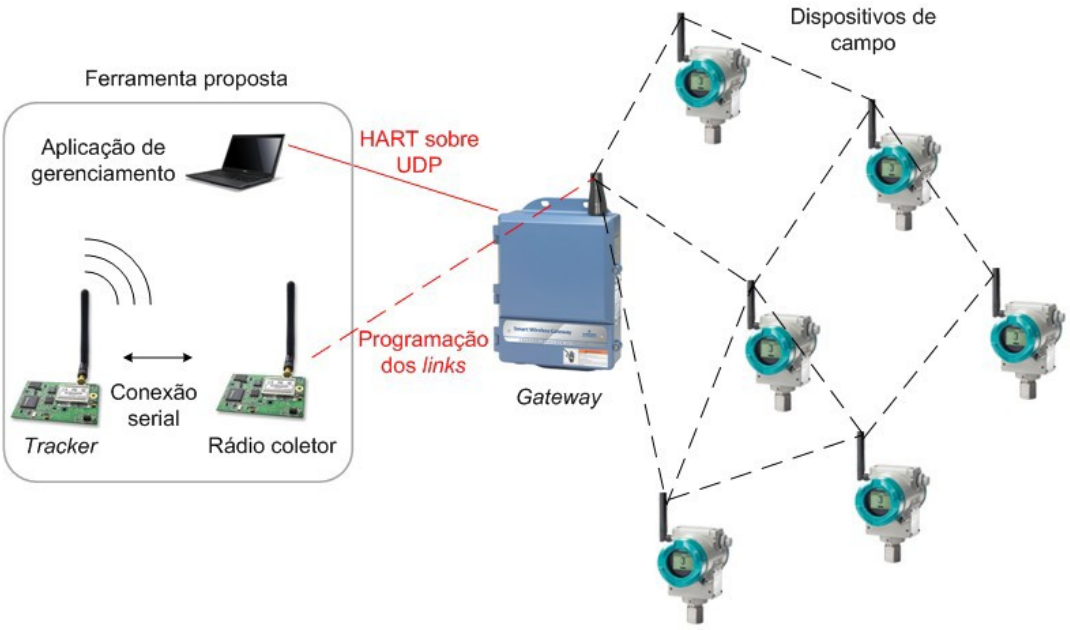


Figura 4 – Ferramenta proposta para análise de redes WH

4.1 APLICAÇÃO DE GERENCIAMENTO

Conforme já descrito no capítulo 2 deste trabalho, o WH faz uso do salto de canais para minimizar as interferências na rede. Dessa forma, a cada *time slot* será utilizado um canal diferente para comunicação. Como a ferramenta proposta faz uso de apenas um rádio coletor,

é preciso que uma aplicação de gerenciamento programe os *links* deste rádio para que a coleta dos dados ocorra nos canais corretos, ou seja, na frequência que de fato esteja ocorrendo comunicação entre os dispositivos da rede.

A aplicação de gerenciamento é um *software* desenvolvido em linguagem C. Tem como objetivo obter os *links* de comunicação de cada um dos vizinhos da unidade coletora e encaminhá-los à própria unidade coletora. Este *software* foi desenvolvido no trabalho de (WINTER, 2011) e foi modificado por (LORENÇATO, 2013).

Esta aplicação é executada em um computador, que deve estar na mesma rede IP (*Internet Protocol* – Protocolo de internet) do *gateway*. A comunicação é realizada através do protocolo HART-IP utilizando UDP (*User Datagram Protocol*).

Para que as mensagens WH capturadas possam ser posteriormente decriptadas pela ferramenta de análise *offline* dos dados, é necessário que a unidade coletora de dados seja capaz de "ouvir" o processo de entrada na rede (*join*) de cada dispositivo, pois é durante este processo que as chaves de sessão da camada de rede são fornecidas pelo gerenciador de rede. Sem estas chaves não é possível decriptar as mensagens. Esse comportamento é comum para qualquer *sniffer* capturando dados em redes WH, inclusive para o *sniffer Wi-Analys*, da HCF.

Dessa forma, a unidade coletora de dados deve ser a primeira a entrar na rede, para que possa capturar todas as chaves de segurança necessárias à medida que os outros dispositivos começarem a ingressar na rede. Assim, após a unidade coletora se unir à rede, a aplicação de gerenciamento deve começar a ser executada ciclicamente e os outros dispositivos podem começar a se unir à rede também.

Com isso, garante-se que a unidade coletora está capturando todos os processos de *join* dos novos dispositivos, e a execução da aplicação de gerenciamento faz com que sejam descobertos os *links* que estes novos dispositivos possuem. Então, gravam-se estes *links* na

unidade coletora, para que esta possa “ouvir” as comunicações dos novos dispositivos, mesmo não participando ativamente destas comunicações.

As etapas que compõem a aplicação de gerenciamento são mostradas na Figura 5 e são descritas a seguir.

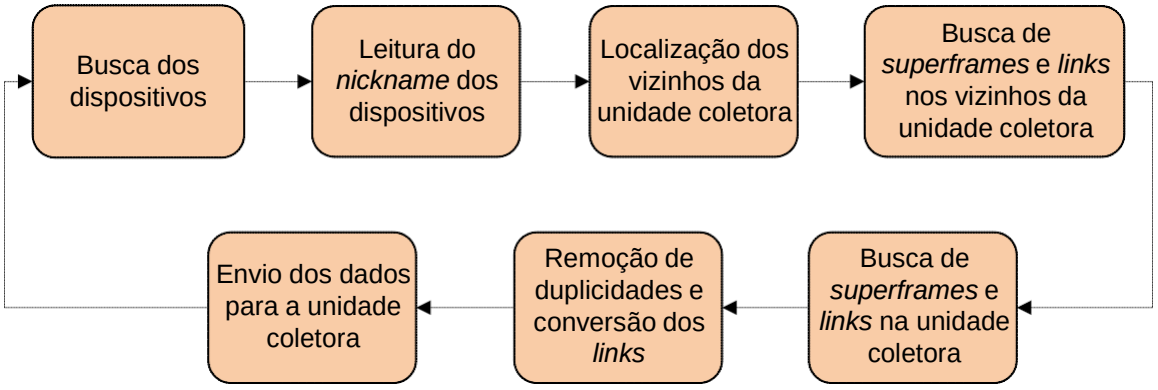


Figura 5 – Etapas da aplicação de gerenciamento

Busca dos dispositivos: nesta etapa, identificam-se todos os dispositivos presentes na rede WH, ou seja, que já se uniram à rede. Para isso, é enviado o comando 814 (*Read Device List Entries* – Leitura das entradas da lista de dispositivos) ao *gateway*. O retorno deste comando é a lista com todos os dispositivos de campo ativos e seus respectivos identificadores únicos (*Unique ID*). Na primeira vez que a aplicação de gerenciamento for executada, somente a unidade coletora pode ser encontrada, pois é o primeiro dispositivo que deve entrar na rede. Se ela ainda não foi encontrada, repete-se esta etapa até que seja detectada.

Leitura do *nickname* dos dispositivos: considerando que a unidade coletora e outro(s) dispositivo(s) já se uniram à rede, então, tem-se uma lista com todos os dispositivos pertencentes à rede. Dessa forma, para cada dispositivo, é enviado ao *gateway* o comando 832 (*Read Network Device Identity using Unique ID* – Leitura da identidade de rede do dispositivo usando o identificador único) passando como parâmetro o identificador único do dispositivo, que já havia sido obtido na etapa anterior. O retorno deste comando é o apelido

(*nickname*) do dispositivo. Assim, tem-se uma associação entre identificador único e apelido. Esta etapa é necessária pois o protocolo HART-IP exige que os comandos sejam enviados utilizando o identificador único, mas as respostas sempre referenciam os dispositivos através dos seus apelidos. Então, esta etapa permite que se possa fazer uma conversão entre os identificadores únicos e os apelidos dos dispositivos.

Localização dos vizinhos da unidade coletora: nesta etapa, são encontrados os vizinhos com *link* e sem *link* com a unidade coletora. Para isso, enviam-se à unidade coletora dois comandos. Um deles é o comando 780 (*Report Neighbor Health List* – Relatório com a lista da "saúde" dos vizinhos) que fornece estatísticas dos vizinhos com os quais a unidade coletora possui *link*. E o outro comando é o 787 (*Report Neighbor Signal Levels* – Relatório com o nível de sinal dos vizinhos), que indica os vizinhos descobertos, mas ainda sem *link* com a unidade coletora. Com o resultado destes dois comandos, é possível saber quais dispositivos a unidade coletora poderá monitorar, ou seja, quais dispositivos estão ao seu alcance.

Busca de *superframes* e *links* nos vizinhos da unidade coletora: nesta etapa, são enviados para cada um dos vizinhos da unidade coletora (com *link* e sem *link*) os comandos 783 (*Read Superframe List* – Leitura da lista de superframes) e 784 (*Read Link List* – Leitura da lista de links), de forma a se obter todos os *superframes* e *links* que devem ser monitorados pela unidade coletora.

Busca de *superframes* e *links* na unidade coletora: similar à etapa anterior, porém, os comandos 783 e 784 são enviados à própria unidade coletora.

Remoção de duplicidades e conversão dos *links*: nesta etapa, é feita uma comparação entre os *links* e *superframes* da unidade coletora e dos seus vizinhos, de forma a remover dados duplicados. Com isso, gera-se uma lista final de *links* e *superframes* a serem escritos na unidade coletora. Também é feita uma conversão no tipo dos *links*, de forma a

transformá-los em *links* de recepção com destino *broadcast*. Isso faz com que a unidade coletora consiga capturar os dados trafegando na rede, pois estará sempre em modo de recepção e irá aceitar a mensagem recebida, pois o destino da mesma está configurado como *broadcast*.

Envio dos dados para a unidade coletora: a última etapa consiste em enviar as informações obtidas na etapa anterior para a unidade coletora utilizando o comando 967 (*Write Link* – Escrita de *link*). Com isso, a unidade coletora é programada para receber todas as mensagens trafegando na rede (dos dispositivos que estejam ao seu alcance).

A aplicação de gerenciamento deve ser executada ciclicamente (atualmente, utiliza-se um período de 30 segundos), para que a lista de vizinhos da unidade coletora seja atualizada e os novos *links* possam ser escritos na mesma. Porém, isso aumenta o tráfego na rede, pois a aplicação gerenciadora faz requisições ao *gateway* que, por sua vez, as transmite aos dispositivos de campo para obter as respostas.

No caso de se estar realizando o monitoramento de uma rede conhecida, ou seja, em que se sabe quais dispositivos deseja-se monitorar, não é necessário executar a aplicação de gerenciamento indefinidamente, pois uma vez que todos os dispositivos sob análise já tenham entrado na rede e seus *links* já tenham sido escritos na unidade coletora, pode-se cancelar a execução da aplicação de gerenciamento, para evitar o tráfego extra gerado pela mesma.

4.2 *HARDWARE* DO RÁDIO COLETOR

O rádio coletor utilizado neste trabalho é mostrado na Figura 6, e foi desenvolvido inicialmente no trabalho de (MULLER et al., 2010). Utiliza o SoC (*System-on-a-chip* – sistema em um chip) MC13224 fabricado pela Freescale. Este componente incorpora um transceptor IEEE 802.15.4 completo na frequência de 2,4 GHz, com baixo consumo de

potência e utiliza um processador ARM7 de 32 bits (ARM7TDMI-S). O transceptor suporta transmissão, recepção, CCA, detecção de energia e medição do LQI, conforme requerido pela norma IEEE 802.15.4 (FREESCALE SEMICONDUCTOR, 2012).

A memória do MC13224 é composta por RAM (96 kbytes), ROM (80 kbytes) e *serial flash* (128 kbytes). A memória ROM inicialmente contém o código de *boot*, *drivers* e a camada MAC 802.15.4. A *serial flash* é acessada durante o *boot* para carregar/inicializar a RAM para a execução do programa. Pode ser espelhada nos 96 kbytes de RAM, sendo que, dessa forma, tem-se um excesso de 32 kbytes para outros códigos além do programa principal. Apenas 4 kbytes são reservados para uso de fábrica, então os 28 kbytes adicionais podem ser usados para armazenamento não volátil de dados.

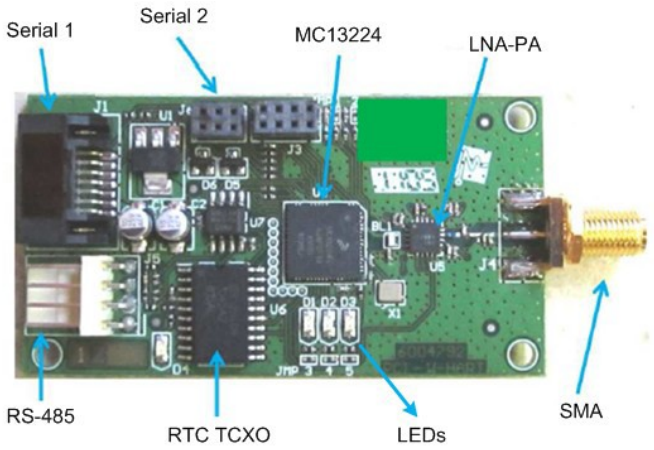


Figura 6 – *Hardware* do rádio coletor

O rádio coletor possui um LNA-PA (*Low noise amplifier e Power amplifier* – amplificador de baixo ruído e amplificador de potência) para aumentar a sensibilidade do receptor em 10 dB e a potência de saída para até 20 dBm (embora o protocolo WH defina a potência de saída em 10 dBm, a norma IEEE 802.15.4 permite até 20 dBm). Também é utilizado um RTC TCXO (*Real-time clock temperature compensated crystal oscillator* – relógio de tempo real com cristal oscilador com compensação de temperatura) para que as restrições temporais do TDMA sejam atendidas, uma vez que o protocolo WH exige um RTC

com tolerância máxima de 3 ppm para que não ocorra a perda de sincronização. Uma antena externa é acoplada ao conector SMA (*SubMiniature Version A*).

O dispositivo também possui duas portas seriais, uma porta RS-485 e LEDs para indicar estados da execução do programa principal.

4.3 *HARDWARE* DO *TRACKER*

O rádio *tracker* utilizado neste trabalho foi desenvolvido no trabalho de (MULLER et al., 2012) e é mostrado na Figura 7.

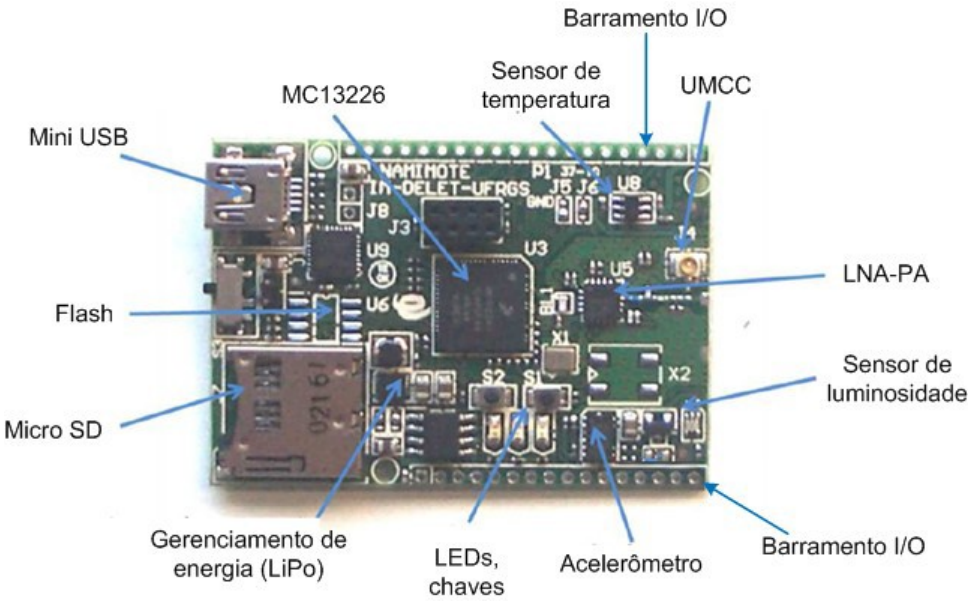


Figura 7 – *Hardware* do *tracker*

Este rádio foi elaborado para utilização em redes de sensores sem fio e, portanto, é primariamente alimentado a bateria. A recarga da bateria é feita por um cabo mini USB. Possui sensores integrados (temperatura, luminosidade e aceleração em três eixos) e gerenciamento de energia para uma bateria de lítio-íon de 3,7 Volts.

Uma característica importante é a presença do suporte para cartão de memória micro SD (*Secure Digital* – segurança digital), que é utilizado neste trabalho para armazenar os dados capturados da rede WH. O dispositivo também possui dois barramentos I/O

(*Input/Output* – entrada/saída) para expansão, sendo um de 16 pinos e outro de 20 pinos. Estes barramentos possuem pinos para módulos de comunicação I^2C (*Inter-Integrated Circuit*), SPI (*Serial Peripheral Interface*), alimentação e outros pinos para uso geral (GPIOs – *General Purpose Input Output*). Alguns dos GPIOs serão utilizados neste trabalho para realizar a comunicação entre o rádio coletor e o *tracker*.

O SoC utilizado no rádio *tracker* é o MC13226, da Freescale. Este componente pertence à mesma família do MC13224 utilizado no rádio coletor, porém é uma versão mais recente. A única diferença entre eles é o conteúdo da memória ROM, que foi modificada no MC13226 para que se tenha mais espaço disponível na RAM para o código da aplicação (FREESCALE SEMICONDUCTOR, 2012).

4.4 INTERAÇÃO ENTRE O RÁDIO COLETOR E O *TRACKER*

O analisador de redes WH proposto neste trabalho necessita de dois rádios conectados fisicamente. O rádio coletor captura as mensagens WH trafegando pela rede e as encaminha ao *tracker*, para que este faça o armazenamento das mensagens no cartão de memória e realize a medição de energia no canal em uso.

O sentido de comunicação entre os dois rádios é sempre do rádio coletor para o *tracker*, sendo que se pode dividir em três tipos possíveis de comunicação/interação entre os rádios:

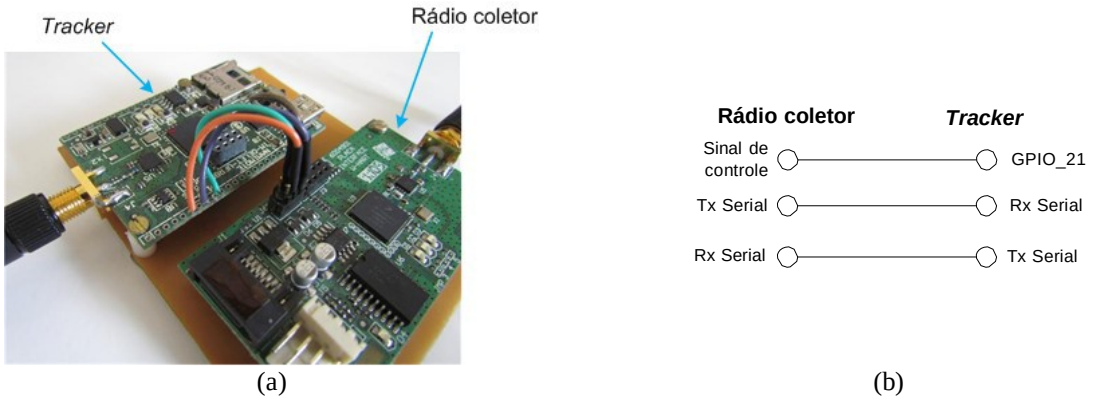
Envio do canal e *Absloute Slot Number* (ASN): o rádio coletor envia a sequência de caracteres “R”+ “CH”+ “ASN”, sendo que o caractere R tem apenas a função de permitir que o *tracker* identifique o tipo de informação que está recebendo, ou seja, para que se possa diferenciar de uma mensagem WH capturada da rede, que não irá iniciar com o caractere R. CH é o canal em que o rádio coletor irá capturar a próxima mensagem WH. É com esta informação que o *tracker* irá realizar a medição de energia no canal correto. E o ASN, que é

uma contagem sequencial do número de *time slots* existentes desde a criação da rede, permite que a ferramenta *offline* de análise dos dados possa fazer uma associação entre as mensagens capturadas e a energia medida nos canais, de forma a apresentar ao usuário, corretamente, a análise de interferências na rede.

Envio da mensagem capturada: todas as mensagens WH que forem capturadas pelo rádio coletor serão enviadas para o *tracker* armazená-las no cartão de memória.

***Set* do sinal de controle indicando o início e o fim da recepção de uma mensagem:** neste caso, não se tem exatamente a transmissão de uma mensagem, mas sim uma sincronização via *hardware* que indica ao *tracker* o momento exato em que a medição de energia deve ser inicializada e finalizada. Para isso, o rádio coletor configura um sinal em nível lógico 1 para indicar o início da recepção da mensagem e, após isso, configura este mesmo pino para o nível lógico 0 para sinalizar o fim da recepção da mensagem. Com isso, o *tracker* consegue medir a energia durante o tempo correto, mesmo com mensagens de tamanhos variados sendo recebidas pelo rádio coletor.

O envio de mensagens do rádio coletor para o *tracker* é feito através de uma conexão serial, enquanto a configuração do sinal de controle é apenas uma conexão física direta de um pino do rádio coletor até um pino do *tracker*. A Figura 8 ilustra a conexão entre os dois rádios.



(a) (b)

Figura 8 – Conexão física entre o rádio coletor e *tracker*. (a) Ilustrativo. (b) Esquemático.

Os SoCs da família MC1322x possuem dois módulos UART (*Universal Asynchronous receiver/transmitter* – transmissor/receptor assíncrono universal). No rádio *tracker*, o UART1 é conectado ao mini USB e o UART2 é usado para conectar a porta serial RS-232. É através do UART2 que é feita a conexão do *tracker* com o rádio coletor.

4.5 O *FIRMWARE* DO RÁDIO COLETOR

O rádio coletor precisa capturar todas as mensagens WH que estiverem ao seu alcance. Como o rádio coletor também é um dispositivo WH capaz de rotear mensagens, então, em alguns casos, a mensagem capturada será simplesmente uma mensagem que possui como origem ou destino o próprio rádio coletor. Já no caso mais complexo, o rádio coletor terá que capturar uma comunicação entre outros dois rádios na rede WH. Isso é possível devido à aplicação de gerenciamento, que programa o rádio coletor para realizar a recepção das mensagens no *time slot* correto. A Figura 9 ilustra os três casos de captura de mensagens pelo rádio coletor.

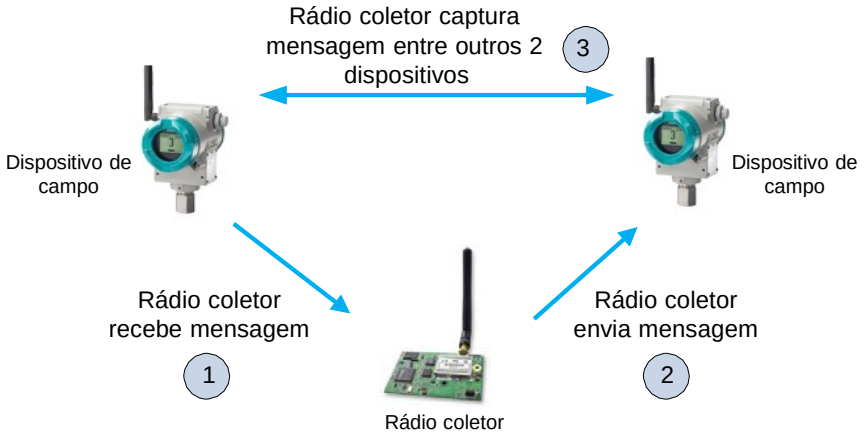


Figura 9 – Mensagens capturadas pelo rádio coletor

O *firmware* do rádio coletor foi desenvolvido a partir de um *firmware* de um dispositivo de campo WH, desenvolvido pelo próprio grupo de pesquisas no âmbito de um projeto de pesquisa em parceria com a Petrobrás, sendo que foram feitas alterações,

principalmente na camada de enlace, para que o rádio coletor tivesse a capacidade de coletar as mensagens nas três situações mostradas na Figura 9. Estas alterações foram desenvolvidas no trabalho de (LORENÇATO, 2013). Com isso, o rádio coletor está apto a enviar pela sua porta serial todas as mensagens WH e ACKs capturados da rede. Além disso, são enviadas também as seguintes informações (juntamente com cada mensagem capturada): *timestamp* da mensagem, ASN, canal em que a captura ocorreu e RSL (*Received Signal Level* – Nível de sinal recebido). Esta estrutura de mensagem será melhor explicada na seção 4.7 deste trabalho.

Além das alterações já desenvolvidas por (LORENÇATO, 2013), foi necessário realizar mais duas modificações no *firmware* do rádio coletor:

- quando o rádio coletor for receber uma mensagem, deve-se enviar pela porta serial a sequência “R”+ “CH”+ “ASN”, para que o *tracker* possa identificá-la e realizar a detecção de energia;
- deve-se manter um sinal de controle em nível lógico alto durante a recepção de uma mensagem, alterando-o para nível lógico baixo após o término da recepção.

4.6 O *FIRMWARE* DO *TRACKER*

O *firmware* do *tracker* foi desenvolvido utilizando um *stack* disponibilizado pela Freescale, denominado de *Simple MAC* (SMAC – *Simple Media Access Controller*). O SMAC é um protocolo desenvolvido em linguagem C baseado no protocolo IEEE 802.15.4 e implementado para funcionar com os transceptores da Freescale. Tem como objetivo ser usado em desenvolvimento rápido de produtos e avaliação de sistemas. Aplicações de baixo custo e que requerem primitivas básicas, como transmissão, recepção e seleção de canais, são bons exemplos de uso do SMAC (HUITRÓN, 2013). Por isso, neste trabalho optou-se por

desenvolver o *firmware* do *tracker* utilizando o SMAC, uma vez que a única funcionalidade de transceptor necessária é a detecção de energia, que é baseada na recepção de sinais.

Basicamente, o SMAC pode ser definido como um *driver* entre o transceptor e a MCU. Além disso, inclui funções para inicializar a MCU e periféricos, tais como: LEDs (*Light Emitting Diodes*), LCD (*Liquid Crystal Display*), entre outros. O código é compilado utilizando o *software* IAR Embedded Workbench IDE. No caso do *tracker*, foi utilizada a versão 5.50.

A Figura 10 apresenta um esquema do *firmware* do *tracker* e a seguir cada uma das etapas é explicada detalhadamente.

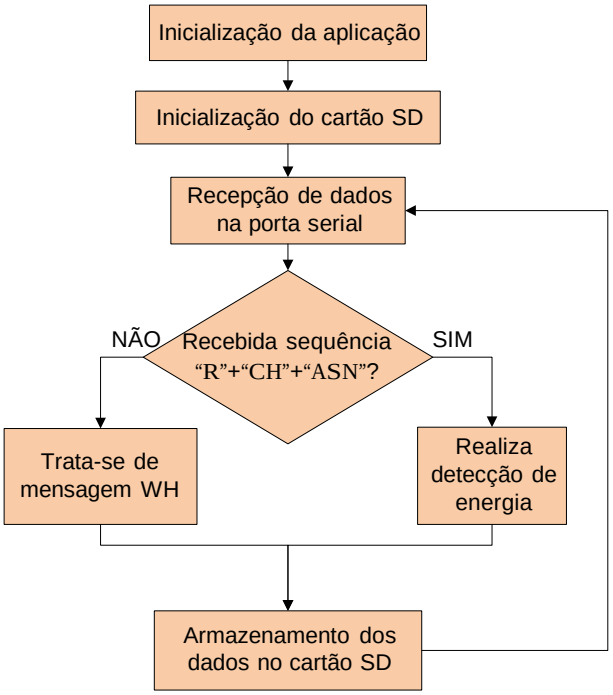


Figura 10 - Esquema do *firmware* do *tracker*

4.6.1 Inicialização da aplicação

Consiste em inicializar elementos tais como: UART, *timers*, interrupções, LEDs, transceptor, entre outros. Grande parte das funções é disponibilizada pelo *stack* SMAC já citado anteriormente.

4.6.2 Inicialização do cartão SD

Os cartões SD são dispositivos de armazenamento removíveis que possuem como características o pequeno tamanho, simplicidade, baixo consumo de potência e baixo custo, o que os torna adequados para diversas aplicações. A especificação da camada física dos cartões SD é definida pela norma referenciada em (SD GROUP, 2013).

Os cartões SD podem se comunicar através de dois modos: *SPI Mode* e *SD Mode*. O *SD Mode* é o modo padrão de transferência do cartão SD, sendo que são transferidos 1 bit ou 4 bits por ciclo de *clock*, dependendo da largura do barramento. O *SPI Mode*, que é o modo utilizado no *firmware* do *tracker*, opera utilizando a interface SPI (*Serial Peripheral Interface*), que é um protocolo serial síncrono para interfacear dispositivos periféricos com microcontroladores. O SPI permite que vários dispositivos troquem dados em *full-duplex*. A vantagem do *SPI Mode* é a possibilidade de utilizar *hosts off-the-shelf* (ex: microcontroladores comerciais com interface SPI), reduzindo o esforço de projeto ao mínimo (SD GROUP, 2013).

A inicialização do cartão SD requer uma sequência específica de comandos, que é descrita na norma definida por (SD GROUP, 2013). A Figura 11 ilustra a sequência utilizada no *firmware* do *tracker*.

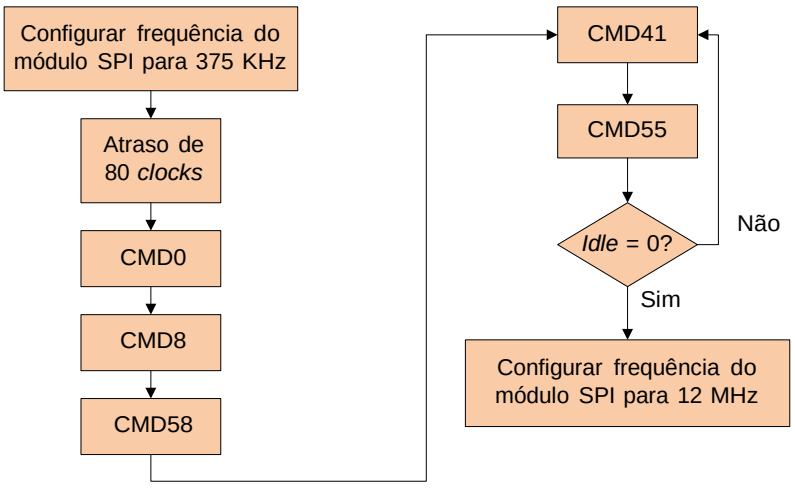


Figura 11 – Etapas da inicialização do cartão SD

Inicialmente, foi preciso modificar algumas configurações do módulo SPI do microcontrolador: o modo de operação foi escolhido como mestre e a frequência de *clock* foi modificada para 375 kHz. A alteração na frequência é requerida para que haja compatibilidade entre uma grande quantidade de cartões SD.

Antes de enviar qualquer comando para o cartão SD, é preciso inicializar os registradores internos do cartão. Para isso, o mestre deve enviar, no mínimo, 80 pulsos de *clock* para o cartão SD, enquanto o sinal CS (*Chip Select*) é mantido em nível alto. Após isso, o cartão passa a responder aos comandos enviados pelo mestre. Enviar pulsos de *clock* significa enviar o valor 0xFF para o cartão SD a cada ciclo de *clock*.

Por padrão, o cartão SD é inicializado no modo SD. Para alterar para o modo SPI, é preciso enviar o comando CMD0 enquanto o sinal CS é mantido em nível lógico baixo. O cartão envia uma resposta indicando que alterou para o modo SPI com sucesso. Após isso, utiliza-se o comando CMD8 para verificar se o cartão SD possui versão 2.00 ou superior e se a tensão aplicada é compatível com o cartão SD. Caso as duas verificações sejam verdadeiras, utiliza-se o comando CMD58 para identificar cartões que não possuam uma faixa de tensão desejada pelo *host*.

Após isso, enviam-se continuamente os comandos CMD41 e CMD55 até que a resposta do cartão indique que a inicialização foi concluída, ou seja, o bit "*in idle state* (em estado inativo)" assuma o valor lógico zero. Por fim, é necessário reconfigurar a frequência de *clock* do módulo SPI do microcontrolador para a máxima permitida que, no caso, é 12 MHz.

A etapa de inicialização do cartão SD no *firmware* do *tracker* ainda inclui a abertura ou criação de um arquivo de texto onde os dados são armazenados. Os cartões SD utilizam o sistema de arquivos FAT (*File Allocation Table* – Tabela de alocação de arquivos). A Tabela 5 mostra os tipos de cartão SD existentes, suas capacidades de armazenamento e o sistema de arquivos utilizado.

Tabela 5 – Tipos de cartões SD e suas especificações

	SD	**SDHC (High Capacity)**	**SDXC (Extended Capacity)**
Capacidade	≤ 2GB	> 2GB, ≤ 32 GB	> 32 GB, ≤ 2TB
Sistema de arquivos	FAT 16	FAT 32	exFAT

De acordo com (SD GROUP, 2013), para cartões SD padrão, os blocos de dados para escrita/leitura podem variar de 1 a 512 bytes. No caso de cartões SDHC e SDXC, o comprimento do bloco é fixado em 512 bytes. Neste trabalho, os testes foram realizados utilizando um cartão SD padrão de 2GB e os dados são agrupados e escritos no cartão SD a cada 512 bytes.

4.6.3 Recepção dos dados na porta serial

Após a inicialização da aplicação e do cartão SD, o *firmware* do *tracker* entra em um *loop* aguardando que os dados sejam recebidos em sua porta serial. Conforme já mencionado, o microcontrolador MC1322x possui *drivers* em sua memória ROM, entre eles o *driver* de leitura/escrita do módulo UART. A documentação de todos estes *drivers* é encontrada em (FREESCALE SEMICONDUCTOR, 2011a).

Uma das funções disponibilizadas pelo *driver* UART é denominada *UartReadData*, que é chamada na inicialização do módulo UART (inicialização da aplicação). Esta função recebe como parâmetros a identificação do módulo UART (uma vez que existem dois módulos), o número de bytes a serem recebidos, um ponteiro para um *buffer* onde serão armazenados os dados lidos e um valor booleano denominado *UartDirectFifoMode*. Se este último parâmetro possuir o valor falso, a função irá esperar pelo número de bytes requisitados e irá retornar. Então, o usuário obtém os valores lidos através do *buffer*.

No caso do último parâmetro possuir o valor verdadeiro, a função irá expor a FIFO (*First In First Out*) de recepção diretamente ao usuário. Neste caso, o número de bytes e o

ponteiro para o *buffer* serão ignorados. Cada módulo UART possui FIFOs de recepção e transmissão independentes com capacidade de 32 bytes cada uma. O conteúdo da FIFO de recepção não será copiado para o *buffer* à medida que os caracteres forem sendo recebidos. Em vez disso, uma *callback* de leitura será chamada se um limiar da FIFO de recepção for atingido (o *threshold* padrão é 5 bytes, mas neste trabalho foi alterado para 2 bytes para atender os requisitos temporais da medição de energia) ou se foi recebido pelo menos um caractere e, após isso, nenhum caractere foi recebido por, no mínimo, 8 períodos de caracteres.

Depois que o *driver* chamou a *callback* de leitura, o usuário precisa ler o conteúdo da FIFO utilizando a função *UartGetByteFromRxBuffer* até que esta retorne falso, indicando que a FIFO está vazia, ou seja, todos os dados foram lidos (FREESCALE SEMICONDUCTOR, 2011a).

O *firmware* do *tracker* utiliza o parâmetro *UartDirectFifoMode* com o valor verdadeiro. A Figura 12 ilustra um esquema da função que foi criada para realizar a leitura dos dados recebidos na serial e a classificação dos dados, ou seja, a diferenciação entre a sequência “R”+ “CH”+ “ASN” e uma mensagem WH capturada da rede.

Variáveis utilizadas:

unsigned int u8TempData[50]: vetor para armazenar dados lidos da serial unsigned int rxDataAnalysis[14]: vetor para armazenar sequência "R+CH+ASN" unsigned int dataCard[512]: vetor para armazenar dados a serem escritos no cartão SD

Fluxograma:

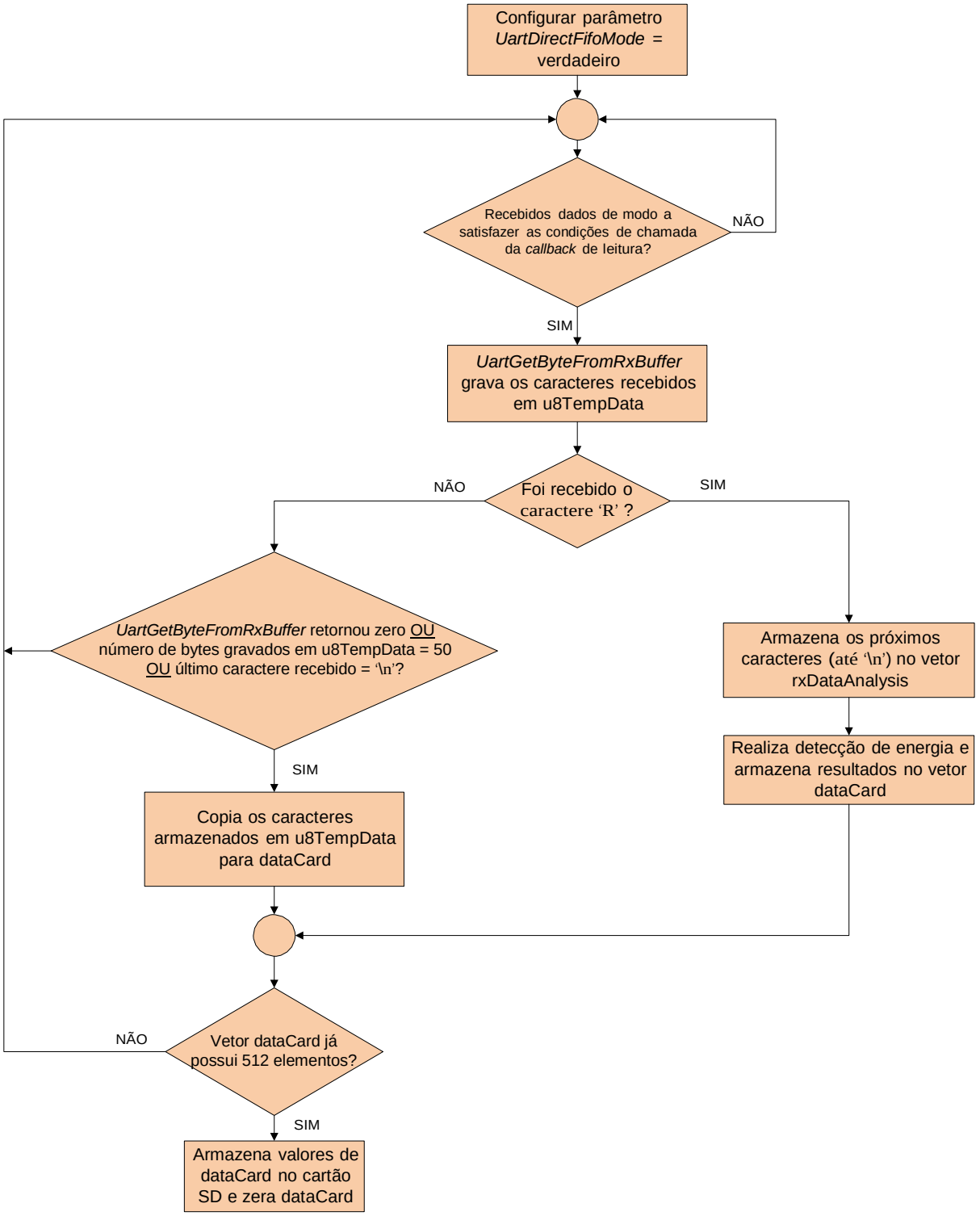


Figura 12 – Esquema da função que realiza a leitura dos dados recebidos na serial

Conforme já mencionado, quando a *callback* de leitura for chamada, significa que foram recebidos dados na serial, e então deve ser chamada a função *UartGetByteFromRxBuffer*. Esta função copia um caractere da FIFO de recepção para um *buffer* fornecido pelo usuário que, no caso, é o vetor denominado *u8TempData*. Como apenas um caractere é copiado por vez, é necessário chamar a função *UartGetByteFromRxBuffer* diversas vezes, até que retorne falso, indicando que todos os dados já foram lidos. Esta repetição é feita utilizando o controle de fluxo *while*. Mas não basta apenas verificar o retorno de *UartGetByteFromRxBuffer* para decidir a permanência no *while*, pois o vetor onde os caracteres lidos são armazenados possui apenas 50 posições. Dessa forma, se todas as posições já tiverem sido preenchidas, é necessário dar continuidade ao fluxo de execução do programa para que os dados deste vetor sejam armazenados no cartão SD e novos dados possam ser lidos da serial, sobrescrevendo os valores do vetor *u8TempData* que já foram tratados.

Um terceiro critério de permanência no *while* tem relação com o caractere '\n', que indica "nova linha". Todas as mensagens enviadas pelo rádio coletor para o *tracker* sempre terminam com '\n'. Dessa forma, quando este caractere é recebido, significa que a mensagem foi totalmente lida da serial e, por isso, deve-se dar continuidade ao fluxo do programa.

Quando o caractere 'R' é recebido, se tem um tratamento diferente, pois este é o caractere que identifica a sequência "R"+ "CH"+ "ASN". Então, quando se identifica o caractere 'R', além de armazenar os dados no vetor *u8TempData*, deve-se também armazená-los no vetor *rxDataAnalysis*. Quando for recebido o caractere '\n', que indica o fim da sequência "R"+ "CH"+ "ASN", executa-se um *break* no *while*, pois a detecção de energia deve ser realizada imediatamente, de forma a existir uma sincronização temporal entre a recepção da mensagem WH pelo rádio coletor e a detecção de energia pelo *tracker*. A detecção de energia será tratada na subseção seguinte.

Por fim, tem-se o uso do vetor *dataCard*, que pode armazenar até 512 caracteres e tem como objetivo agrupar os dados antes de serem armazenados no cartão SD. Se uma das três condições de saída do *while* for satisfeita, os caracteres armazenados em *u8TempData* serão copiados para *dataCard*. Do mesmo modo, quando a detecção de energia for realizada, os resultados também serão armazenados em *dataCard*. Quando as 512 posições deste vetor já tiverem sido preenchidas, estes dados são copiados definitivamente para o cartão SD, e este vetor pode ser sobrescrito com novos valores.

4.6.4 Detecção de energia

Após toda a sequência “R”+ “CH”+ “ASN” ter sido recebida, imediatamente será chamada uma função que realiza a detecção de energia, e que receberá o vetor *rxDataAnalysis* como parâmetro. A Figura 13 mostra um esquema da função que realiza a detecção de energia.

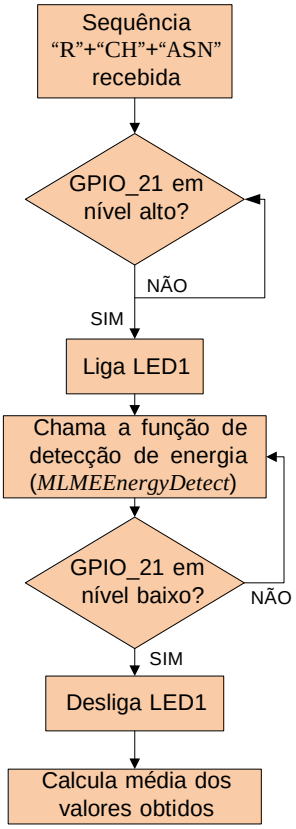


Figura 13 - Esquema da função que realiza a detecção de energia

Conforme já mencionado anteriormente, existe uma sincronização via *hardware* entre o rádio coletor e o *tracker*, que é feita por um sinal controlado pelo rádio coletor e que é recebido pelo *tracker* no GPIO 21. Trata-se de uma ligação física entre os dois rádios.

Quando o rádio coletor inicia a recepção de uma mensagem WH, ele altera este sinal de controle para nível lógico 1, e quando a recepção for finalizada, o sinal é alterado para nível lógico zero. Esta sincronização entre os dois rádios é importante, pois o *tracker* não tem como "conhecer" o tempo correto de recepção da mensagem, uma vez que as mensagens WH podem ter tamanhos variáveis, dependendo do número de bytes da mensagem, sendo limitado em 4256 microssegundos. Assim, esta sincronização permite que o *tracker* faça a medição de energia somente durante o tempo em que a mensagem está de fato sendo recebida pelo rádio coletor.

De acordo com a Figura 13, a função que realiza a detecção de energia ficará aguardando o GPIO 21 estar em nível lógico alto. Quando isto ocorrer, o LED 1 será ligado e será chamada uma função disponibilizada pelo SMAC que, de fato, realiza a detecção de energia no canal desejado. Enquanto o GPIO 21 não for alterado para nível lógico zero, a detecção de energia continuará sendo executada. Após isso, o LED 1 será desligado e será feita uma média de todos os valores de energia medidos.

A função disponibilizada pelo SMAC é denominada *MLMEEnergyDetect* e está documentada em (FREESCALE SEMICONDUCTOR, 2011b). Essa função recebe como parâmetro o canal onde a energia deve ser medida, que é obtido do vetor *rxDataAnalysis.*

O que essa função faz é realizar um ciclo de CCA (avaliação de canal livre), retornando o valor da energia medida, que é um número de 8 bits e, portanto, está dentro da seguinte faixa:

0x00: praticamente não existe atividade no canal

0xFF: o canal tem alta energia e está muito ocupado

A duração do CCA, de acordo com (IEEE, 2011), é 128 microssegundos. Como as mensagens WH podem ter duração de até 4256 microssegundos, é necessário realizar diversas medições de energia, de forma a cobrir todo o tempo de recepção da mensagem. Por isso, é feita uma média de todos os valores de energia obtidos.

Ainda de acordo com (IEEE, 2011), os valores mínimos e máximos de energia (0x00 e 0xFF) devem ser associados com os sinais de menor e maior qualidade detectáveis pelo receptor (-100 dBm e -15 dBm, respectivamente), e quaisquer outros valores devem ser distribuídos uniformemente entre estes dois limites. Então, a função de detecção de energia faz a conversão do valor decimal de energia medido para valores em dBm, utilizando a equação (1):

$$ED_{(dBm)} = \frac{ED_{(dec)}}{3} - 100 \quad (1)$$

Após a medição de energia, os resultados são armazenados no vetor *dataCard* para posterior armazenamento no cartão SD. O seguinte padrão é utilizado: “E”+ “CH”+ “dBm”+ “ASN”, onde “E” é apenas um caractere para indicar que se trata dos dados de detecção de energia, CH é o canal onde a energia foi medida, dBm é o resultado da potência medida e o ASN tem como objetivo permitir a associação dos dados pela aplicação *offline* de análise dos dados.

4.7 FERRAMENTA PARA ANÁLISE DOS DADOS CAPTURADOS

No trabalho de (KUNZEL, 2012) foi desenvolvido um ambiente para avaliação de algoritmos de roteamento para redes WH. Este ambiente é uma aplicação desenvolvida em Java e foi utilizado como base para o desenvolvimento da ferramenta de análise dos dados capturados pelo coletor de dados e *tracker*, os quais estão armazenados no cartão SD. As implementações foram realizadas utilizando o *software* NetBeans IDE 7.3.

Para que os algoritmos de roteamento possam ser avaliados, é necessário que o usuário informe a topologia da rede, ou seja, o número de dispositivos de campo, distância entre os dispositivos, entre outras informações. A ferramenta permite que o usuário informe manualmente estas informações ou utilize um arquivo com dados capturados pelo *sniffer Wi-Analys*, sendo que, neste caso, a topologia da rede é montada a partir dos dados contidos no arquivo. A Figura 14 ilustra a tela inicial do programa, que solicita ao usuário o modo de entrada dos dados: criação manual da rede ou a partir de dados capturados pelo *sniffer*.

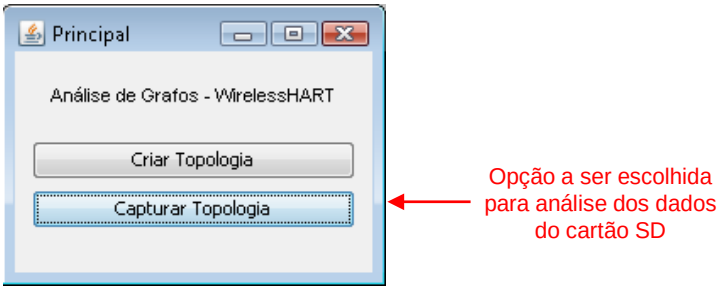


Figura 14 – Tela inicial da ferramenta de análise dos dados

No caso do analisador de redes proposto neste trabalho, os dados capturados encontram-se em um arquivo no cartão SD. Logo, será feito uso de toda a infraestrutura de leitura de arquivo já desenvolvida por (KUNZEL, 2012), realizando modificações para atender ao formato de arquivo gerado pelo *tracker*, que é diferente do arquivo gerado pelo *Wi-Analys*. Além da leitura do arquivo, serão criadas novas telas para apresentar as informações extraídas dos dados coletados.

A Figura 15 apresenta a tela para abertura do arquivo e análise dos resultados. As duas partes indicadas por setas mostram o que foi incluído na tela para implementar a análise dos dados armazenados no cartão SD. Como o *tracker* utiliza um formato de arquivo diferente do que é gerado pelo *Wi-Analys*, foi necessário permitir ao usuário que selecione qual tipo de arquivo deseja abrir. Dessa forma, mantém-se inalterado o tratamento dos dados do *Wi-Analys*

e cria-se um novo tratamento para os dados gerados pelo *tracker*. Também foi incluído um novo botão que apresentará ao usuário as análises e estatísticas dos dados lidos do cartão SD.

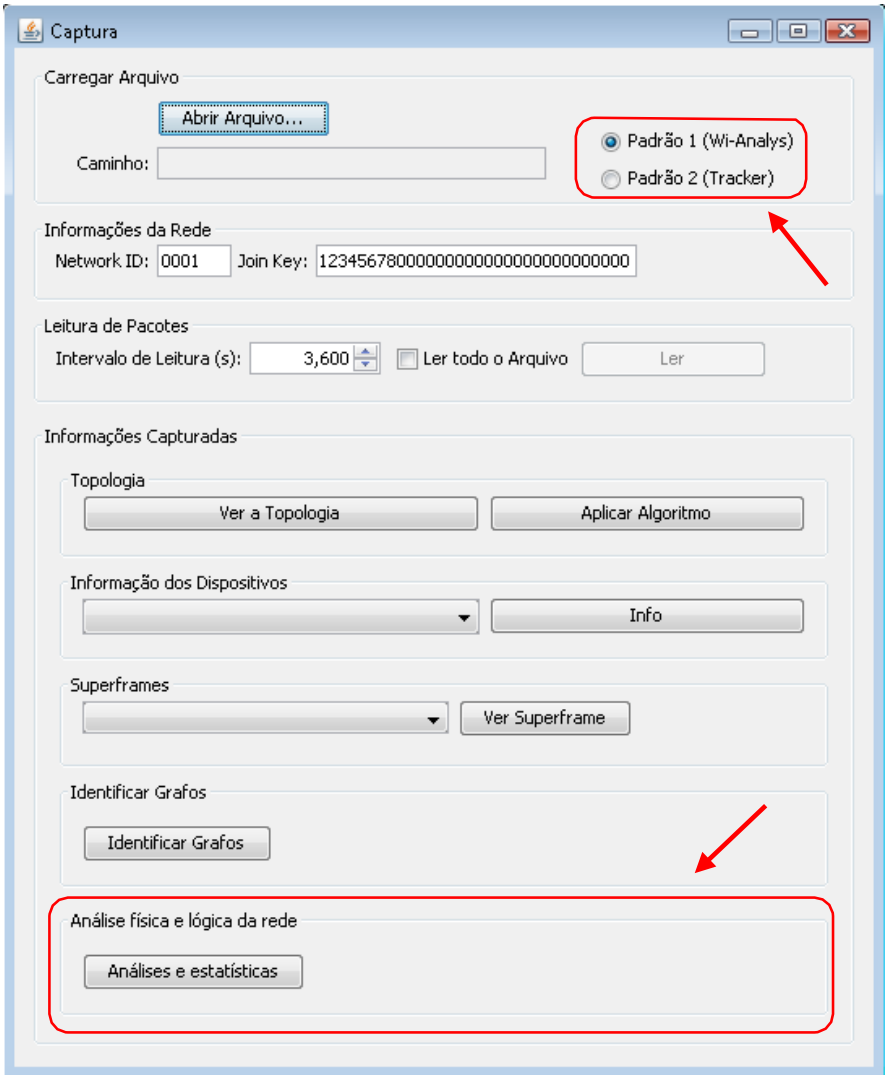


Figura 15 - Tela para abrir arquivo com dados capturados da rede

Os detalhes de implementação das partes não destacadas na Figura 15 não serão abordados neste trabalho, pois o objetivo é focar nas alterações que foram feitas para realizar a análise dos dados gerados pelo *tracker*. Pode-se dividir estas alterações em duas partes: classificação dos dados contidos no arquivo e geração de análises e estatísticas.

4.7.1 Classificação dos dados

O padrão do arquivo gerado pelo *sniffer Wi-Analys* é mostrado no exemplo da Figura 16.

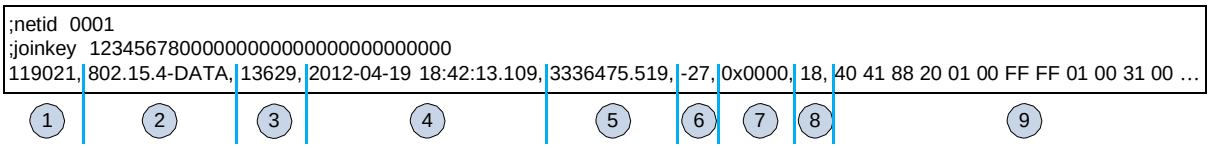


Figura 16 – Padrão dos dados gerado pelo *Wi-Analys*

Os dados são separados por vírgulas e são compostos por nove informações: identificação do *sniffer* (1), protocolo da rede (2), número do pacote capturado (3), horário de captura (4), tempo decorrido (5), RSL (6), status do pacote (checagem do CRC) (7), canal utilizado (8) e mensagem criptografada (9). Dessa forma, a aplicação desenvolvida no trabalho de (KUNZEL, 2012) realiza o tratamento dos dados a partir deste padrão. Devido a isto, o rádio coletor transmite dados para o *tracker* utilizando um padrão similar a este, de modo a facilitar o tratamento dos mesmos pela ferramenta de análise.

Os quatro primeiros elementos mostrados na Figura 16 não impactam na análise dos dados e, portanto, o rádio coletor não irá enviá-los ao *tracker*. No lugar destes quatro elementos, será enviado o ASN, que é necessário para realizar a associação com os dados relativos à medição de energia. A Figura 17 ilustra um exemplo do padrão de dados enviado pelo rádio coletor ao *tracker*.

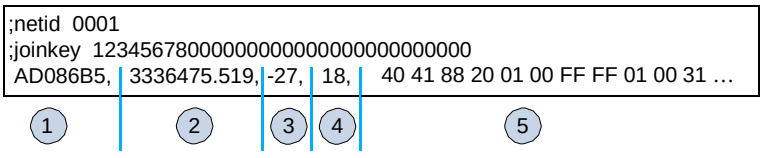


Figura 17 – Padrão de dados enviado pelo rádio coletor

O fato de não ser utilizado um padrão exatamente igual ao do *sniffer Wi-Analys* é explicado pela necessidade de envio do ASN, que não é utilizado no *Wi-Analys*, mas é preciso

que o rádio coletor envie esta informação para que posteriormente a ferramenta de análise dos dados possa fazer uma associação entre as mensagens WH e a energia medida.

O arquivo armazenado no cartão SD contém, além das mensagens WH capturadas pelo rádio coletor (padrão da Figura 17), as informações da detecção de energia e também a sequência “R”+ “CH”+ “ASN”. Então, um exemplo de informações que estarão armazenadas no cartão SD e que deverão ser tratadas pela ferramenta de análise pode ser visto na Figura 18.

```
;netid  0001
;joinkey  12345678000000000000000000000000
R18AD086B5
E, 18, -27, AD086B5
AD086B5, 3336475.519, -27, 18, 40 41 88 20 01 00 FF FF 01 00 31 …
R11AD0862D
E, 11,  -27, AD0862D
AD0862D, 3336475.519, -35, 11, 40 41 88 20 01 00 FF FF 01 00 31 …
```

Figura 18 – Padrão dos dados gravados no cartão SD

Com relação à classificação dos dados contidos no cartão SD (Figura 18), a ferramenta de análise considera, de maneira geral, duas classes para armazenar os dados que são lidos do arquivo: *InputPacket* e *InputPacketTracker*. A Figura 19 mostra os atributos que cada objeto desta classe possui.

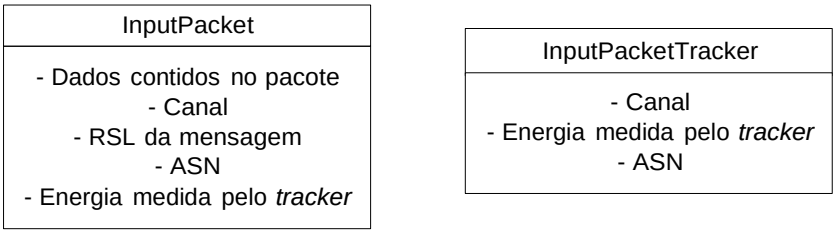


Figura 19 – Classes para classificar e armazenar os dados lidos do arquivo

O atributo ASN foi incluído na classe *InputPacket*, pois esta classe também será utilizada para armazenar os dados do rádio coletor, além dos dados do *sniffer Wi-Analys*.

Um vetor pertencente à classe *InputPacket* armazena os dados originados do *sniffer Wi-Analys* ou as mensagens WH que o rádio coletor enviou ao *tracker*. E um vetor da classe *InputPacketTracker* armazena os dados referentes à detecção de energia.

De acordo com a Figura 18, o cartão SD possui três tipos de mensagens armazenadas, sendo que duas delas são classificadas e armazenadas nos respectivos objetos e o outro tipo de mensagem, que é a sequência “R”+ “CH”+ “ASN”, não é considerada na ferramenta de análise dos dados e, portanto, não é classificada.

A Figura 20 mostra como é feita a classificação dos dados contidos no cartão SD, que é a primeira etapa para realizar a análise dos dados, ou seja, é a primeira ação realizada pela ferramenta quando o usuário utilizar o botão “Ler” na Figura 15.

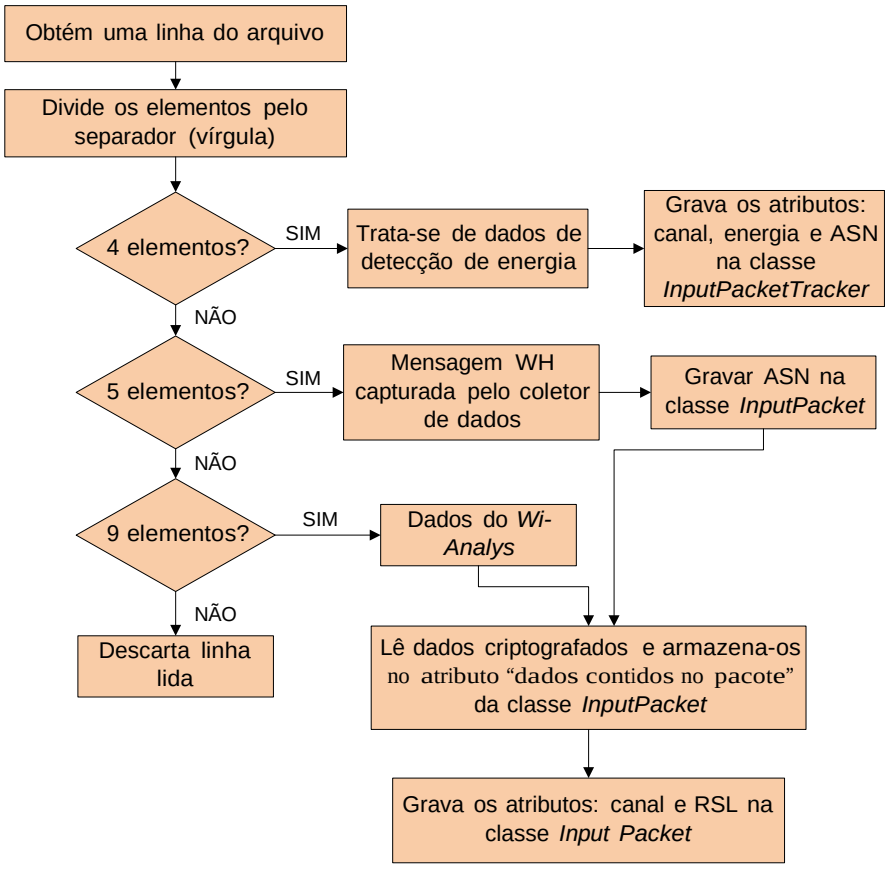


Figura 20 – Classificação dos dados do arquivo

Para poder armazenar corretamente as informações nos vetores das classes *InputPacket* e *InputPacketTracker*, é preciso diferenciar os dados que estão contidos no arquivo. Para isso, cada linha do arquivo é dividida pelo separador de elementos, que é a vírgula. Então, os dados são diferenciados pelo número de elementos, ou seja, se a divisão resultou em nove elementos, isto indica que se trata de dados do *Wi-Analys*. No caso de 5 elementos, indica uma mensagem WH capturada pelo coletor de dados e no caso da linha ser composta por apenas quatro elementos, então se trata de dados referentes à detecção de energia. Neste último caso, basta armazenar os valores do canal, energia medida e ASN no vetor da classe *InputPacketTracker*.

A diferença que existe no tratamento dos dados do *Wi-Analys* e do coletor de dados é referente ao ASN. No caso do coletor de dados, o ASN é enviado no primeiro elemento da linha de dados, conforme mostrado na Figura 17, e deve ser armazenado. Após essa diferenciação, o comportamento dos dois padrões segue o mesmo, ou seja, é feito o armazenamento da mensagem WH, canal e RSL no vetor da classe *InputPacket*.

Após a separação dos dados do arquivo nos dois tipos de mensagens, é necessário fazer uma associação entre os dois tipos de dados. Isso é feito pelo uso do ASN, da seguinte forma: para cada mensagem WH que foi capturada e que está armazenada no vetor da classe *InputPacket*, deve-se fazer uma busca no vetor *InputPacketTracker*, de forma a encontrar o mesmo ASN. Quando for encontrado, obtém-se o valor da energia medida para este ASN e grava-se este valor no atributo “Energia medida pelo *tracker*” no vetor da classe *InputPacket*.

O ASN tem este importante papel de permitir associar os dois tipos de dados, ou seja, permite fazer a correta comparação entre o RSL de uma mensagem e o valor de energia medido durante a captura desta mesma mensagem, fazendo com que seja possível chegar a conclusões a respeito de interferências na rede, o que será abordado no decorrer deste trabalho.

Então, até esta etapa de execução da ferramenta, já se tem todos os dados necessários para gerar as análises e estatísticas, sendo que estão armazenados no vetor da classe *InputPacket*. A próxima etapa é utilizar o botão “Análises e estatísticas” na Figura 15, e obter as análises da rede, conforme descrito na próxima subseção.

4.7.2 Geração de análises e estatísticas

Após os dados do cartão SD terem sido lidos e classificados, a ferramenta de análise irá apresentar ao usuário as análises e estatísticas extraídas dos dados. Para isso, utiliza-se o botão “Análises e estatísticas” mostrado na Figura 15. Será aberta a tela mostrada na Figura 21.

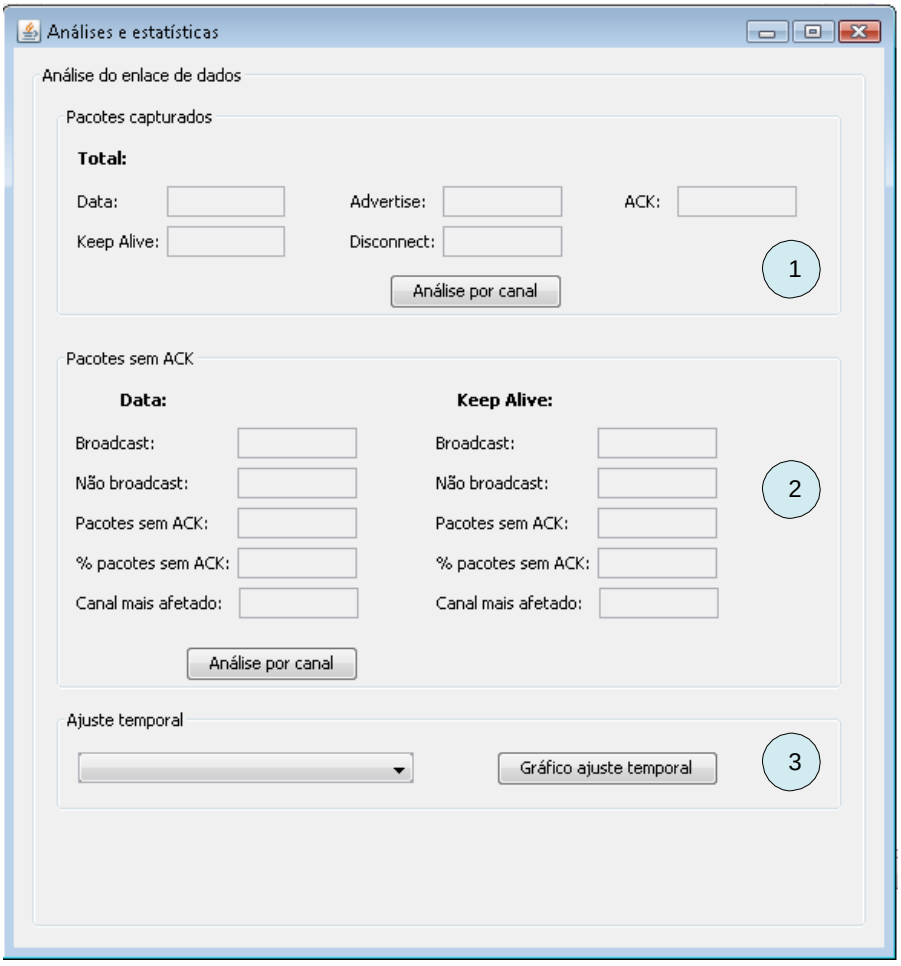


Figura 21 – Tela de apresentação das análises e estatísticas

Os resultados apresentados podem ser divididos em três categorias, que estão numeradas na Figura 21:

1) classificação dos dados por tipo de mensagem WH;

2) contabilização dos pacotes com perda de ACK;

3) ajuste de tempo entre os dispositivos;

4.7.2.1 Classificação dos dados por tipo de mensagem WH

Esta categoria de resultados está indicada pela numeração “1” na Figura 21. Todas as mensagens WH são classificadas de acordo com o seu tipo: *Data*, *Keep Alive*, *Advertise*, *Disconnect* e *ACK*. Com isso, apresenta-se o número total de mensagens WH capturadas e o número de mensagens WH de cada tipo.

Além disso, o botão “Análise por canal” mostra ao usuário a distribuição das mensagens capturadas por canal, conforme mostrado no exemplo da Figura 22 (exemplo ilustrativo). Com isso, consegue-se verificar quais os canais mais utilizados e os canais com nenhuma utilização, que provavelmente devem estar na *blacklist* do gerenciador de rede, que é a lista que contém os canais que não podem ser usados durante o funcionamento da rede. Esta lista é informada pelo próprio usuário da rede e permite a este que elimine alguns canais que podem ser conhecidos por possuírem interferências, por exemplo.

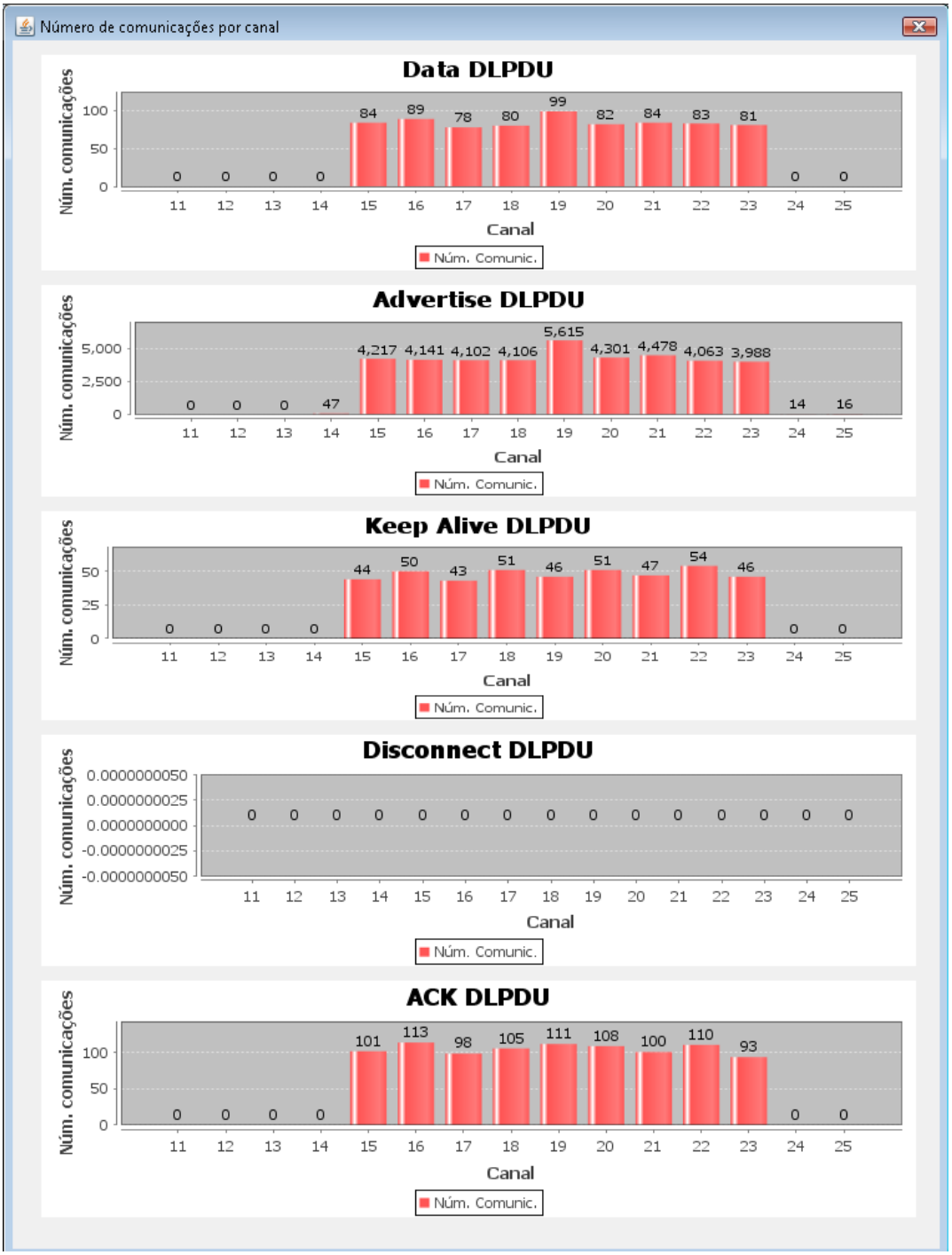


Figura 22 – Distribuição das mensagens WH por canal

4.7.2.2 Contabilização dos pacotes com perda de ACK

Esta categoria de resultados está indicada pela numeração “2” na Figura 21 e tem como objetivo informar ao usuário quantos pacotes tiveram perda de ACK e em quais canais estas perdas ocorreram.

Apenas os pacotes WH do tipo *Data* e *Keep Alive* possuem ACK associado. Porém, se o destino destas mensagens for *broadcast*, não existe ACK associado. Então, a ferramenta de análise mostra ao usuário quantos pacotes são do tipo *Data* e *Keep Alive*, e separa este total de pacotes em *broadcast* e não *broadcast*. Após isso, para cada um dos dois tipos de mensagens, calcula a porcentagem de pacotes sem ACK, utilizando o número de pacotes não *broadcast* e o total de pacotes sem ACK.

A ferramenta também mostra uma importante informação que é a indicação do canal mais afetado, ou seja, o canal que mais apresentou perda de ACK. Com isso, o usuário pode tomar a decisão de incluir este canal na *blacklist* do gerenciador de rede. Além de mostrar o canal com maior perda de ACK, a ferramenta também mostra a distribuição das perdas de ACK por canal, fornecendo ao usuário uma visão geral de todos os canais WH, como pode ser visto no exemplo da Figura 23.

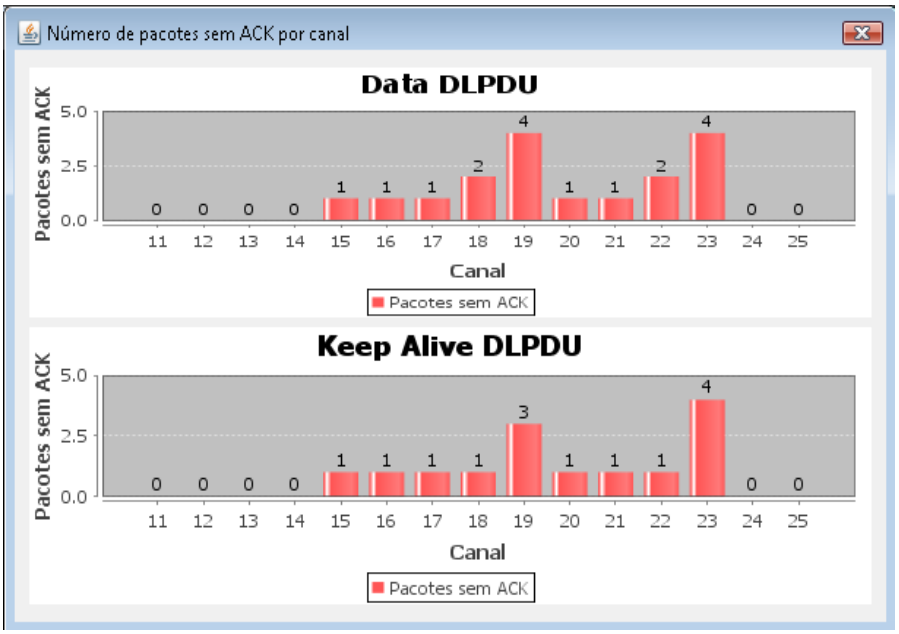


Figura 23 – Distribuição da perda de ACK por canal

4.7.2.3 Ajuste de tempo entre os dispositivos

Esta categoria de resultados está indicada pela numeração “3” na Figura 21 e apresenta as diferenças temporais entre dois dispositivos. Conforme explicado no capítulo 2

deste trabalho, as mensagens do tipo ACK possuem no seu *payload* um parâmetro denominado “Ajuste de tempo”, que é a diferença entre o tempo esperado para o início da recepção de um pacote e o tempo real em que isso ocorreu. É medido em microsegundos, e pode ser positivo, se o pacote foi recebido antes do esperado, ou negativo, se foi recebido depois do esperado.

Então, a ferramenta de análise agrupa todos os pares de dispositivos que tiveram comunicação entre si e analisa as mensagens trocadas entre eles, de forma a obter todos os ajustes de tempo medidos entre estes dois dispositivos. Assim, o usuário seleciona uma combinação de dois dispositivos e é mostrado o gráfico do ajuste temporal entre ambos, tão bem como a média do ajuste de tempo, conforme pode ser visto no exemplo da Figura 24 e Figura 25.

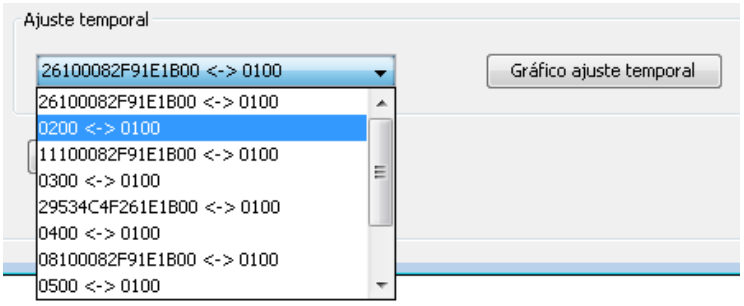


Figura 24 – Tela para seleção da combinação de dois dispositivos

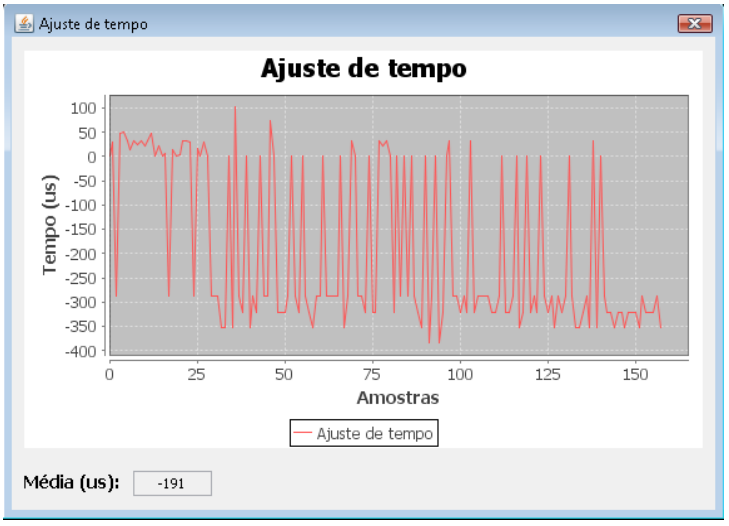


Figura 25 – Gráfico do ajuste de tempo entre dois dispositivos

5 VALIDAÇÃO EXPERIMENTAL E RESULTADOS

A fim de validar a ferramenta proposta e apresentar os resultados obtidos neste trabalho, foram realizados experimentos no Laboratório de Sistemas de Controle, Automação e Robótica (LASCAR) da UFRGS. Foi montada uma rede WH para realizar as seguintes validações experimentais: verificar que a medição de energia está sendo realizada no momento correto, ou seja, durante a recepção de mensagens pelo rádio coletor; garantir que a ferramenta proposta está detectando interferências na rede; e garantir que todas as informações estão sendo capturadas corretamente da rede e que a ferramenta de análise de dados *offline* interpreta os dados e apresenta as estatísticas e análises da rede de forma correta.

5.1 JANELA DE MEDIÇÃO DE ENERGIA

Conforme descrito na seção 4.4, o rádio coletor envia três tipos de dados para o *tracker*: as mensagens WH capturadas da rede, a sequência “R”+ “CH”+ “ASN” e o *set* do sinal de controle indicando o início e o fim da recepção de uma mensagem WH.

O *tracker* precisa realizar a medição de energia durante o tempo em que uma mensagem estiver sendo recebida pelo rádio coletor, para que se possa comparar a energia da mensagem recebida (RSL) com a energia medida no canal. Com isso, consegue-se chegar a conclusões a respeito de possíveis interferências na rede. Dessa forma, uma das validações experimentais da ferramenta proposta é verificar se a medição de energia está sendo realizada, de fato, no momento correto.

De acordo com a seção 4.6, o *tracker* constantemente faz a leitura dos dados da sua porta serial e armazena-os no cartão de memória. Porém, quando é recebido o caractere ‘R’ significa que uma mensagem será recebida em seguida pelo rádio coletor e, portanto, a medição de energia deverá ser realizada. Então, após toda a sequência “R”+ “CH”+ “ASN” ser completamente recebida, o *tracker* irá verificar se o sinal de controle está em nível lógico

1 (o sinal de controle é setado para 1 exatamente no início da recepção da mensagem). Se estiver em nível 1, a medição de energia pode ser realizada, até que o pino volte para o nível lógico zero, indicando o fim da recepção da mensagem.

Foram feitas medições com o osciloscópio a fim de verificar se o tempo de medição de energia pelo *tracker* estava coerente com o tempo de recepção da mensagem. Os resultados são mostrados na Figura 26.

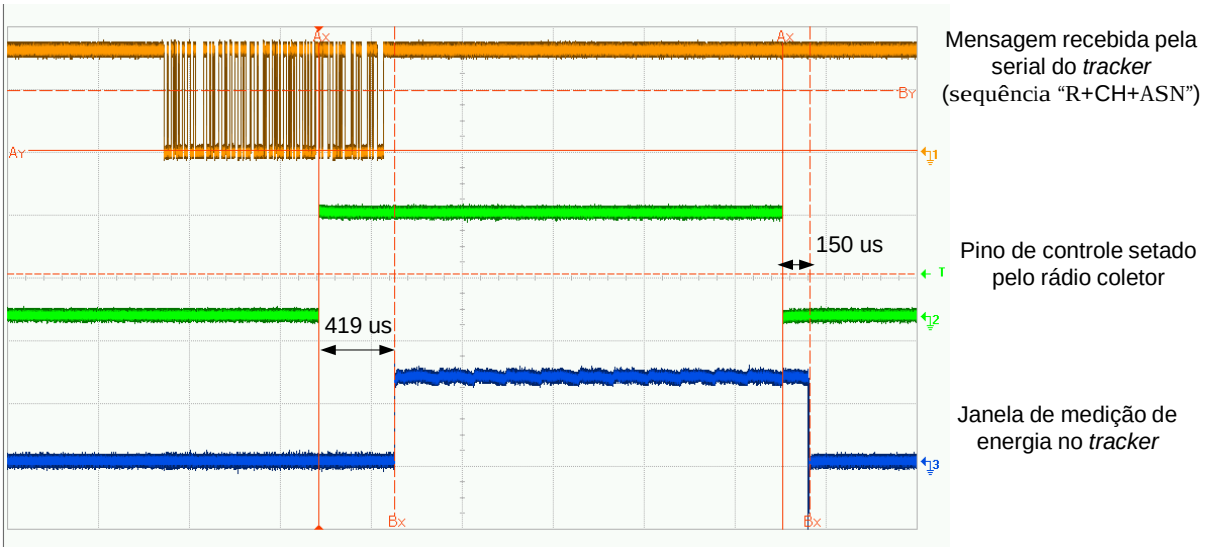


Figura 26 – Análise temporal da janela de medição de energia

As três medições indicadas na Figura 26 foram feitas em pontos específicos no *hardware* do *tracker*. A primeira delas é referente a mensagens sendo recebidas na serial, ou seja, a medição foi feita em um dos fios que interliga o *tracker* ao rádio coletor, por onde passam as mensagens sendo enviadas pelo rádio coletor. Neste caso, está sendo mostrada a recepção da sequência "R"+ "CH"+ "ASN".

A segunda medição é referente ao sinal de controle que faz com que o rádio coletor indique ao *tracker* a duração exata de recepção de uma mensagem. Esta medição foi feita no fio que transmite este sinal do rádio coletor ao *tracker*. E a terceira medição mostra o tempo em que o *tracker* realizou a detecção de energia. Esta medição é feita em um LED no *tracker*,

que é ligado no início da detecção da energia e desligado no final, conforme foi mostrado na Figura 13.

Pela Figura 26, pode-se ver que o início da medição de energia pelo *tracker* ocorre 419 microsegundos após o sinal de controle ter sido alterado para nível lógico 1, ou seja, durante este período, a energia não foi medida, mesmo que a mensagem já estivesse sendo capturada pelo rádio coletor. Isso ocorre, pois a detecção de energia só inicia após o *tracker* ter recebido toda a sequência “R”+ “CH”+ “ASN” e o sinal de controle estar em nível lógico 1. E como se pode perceber, a sequência “R”+ “CH”+ “ASN” começa a ser enviada pelo rádio coletor antes do *set* do sinal de controle, entretanto, só termina depois que o sinal de controle já foi setado para nível 1.

Não há muitas opções para reduzir esse atraso no início da medição de energia. Uma das alternativas seria enviar a sequência “R”+ “CH”+ “ASN” um pouco antes, para fazer com que todos os caracteres sejam recebidos pelo *tracker* antes do *set* do sinal de controle. Mas não é possível fazer isso, pois o envio destes caracteres já está no melhor instante temporal possível.

Outra opção seria reduzir o número de caracteres enviados pelo rádio coletor. Todavia todas as informações são essenciais para o funcionamento geral da ferramenta proposta: o “R” tem a finalidade de diferenciar a sequência “R”+ “CH”+ “ASN” de mensagems WH capturadas. O canal é fundamental para o *tracker* medir a energia no canal correto e o ASN também é necessário para fazer a associação dos dados.

Outra alternativa seria fazer com que o *tracker* não precisasse aguardar todo o recebimento da sequência “R”+ “CH”+ “ASN” para dar início à medição de energia. Em vez disso, aguardaria somente a recepção do “R+CH”, uma vez que somente o canal é necessário para realizar a detecção de energia. E após a medição de energia, o *tracker* deveria ler o restante dos caracteres, que seria o ASN. Ou ainda, o número do canal poderia ser transmitido

ao *tracker* utilizando sinais, juntamente com o *set* do sinal de controle (ao invés de transmitir o número do canal pela porta serial). Esta opção não será testada neste trabalho, mas pode ser analisada e validada em trabalhos futuros.

Uma vez que a duração máxima de uma mensagem WH é 4256 us, então, considerando um atraso de 419 us no início da medição de energia, conclui-se que a energia não é medida durante 9,8% do tempo total de recepção da mensagem.

Pela Figura 26, também se pode observar que o *tracker* finaliza a detecção de energia 150 us após o sinal de controle ter sido alterado para nível lógico zero, que é o tempo que o *firmware* do *tracker* demora para fazer a leitura do sinal de controle e perceber que a medição de energia deve ser finalizada.

5.2 DETECÇÃO DE INTERFERÊNCIAS NA REDE

O objetivo desta validação experimental é verificar que a ferramenta proposta detecta interferências na rede. Para inserir uma interferência de forma controlada, isto é, em um canal específico, foi utilizado o analisador de RF *FieldFox N9912A*, da *Agilent Technologies*. Além disso, foram utilizados os seguintes elementos: rádio coletor e *tracker*, e gerenciador de rede, ponto de acesso e *gateway* da *Emerson* (modelo 1420A), conforme mostrado na Figura 27.

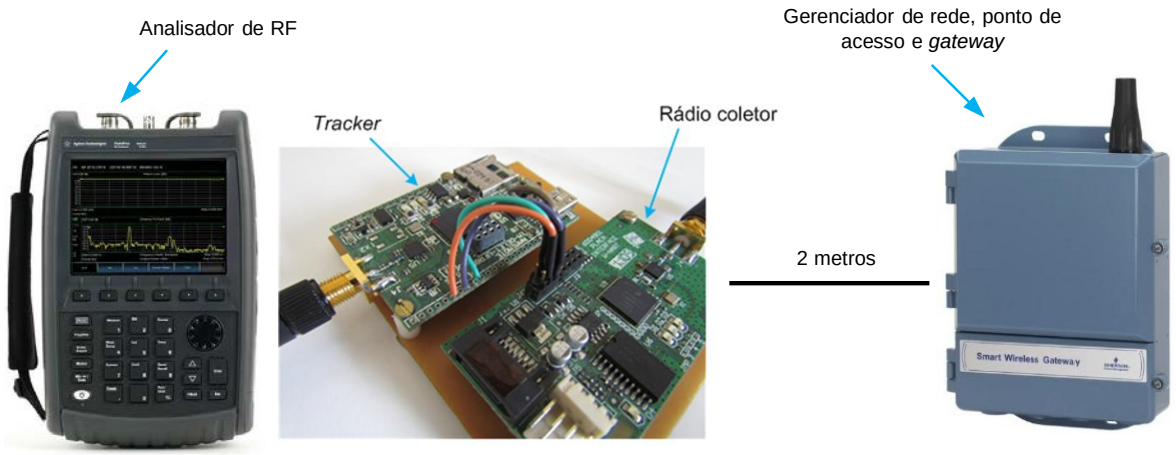


Figura 27 – Elementos utilizados no teste de medição de interferência

A interferência inserida com o analisador de RF *FieldFox N9912A* é um sinal de onda contínua com largura de banda de 2 MHz. Foi utilizada atenuação de 0 dB, o que resulta em uma saída entre -2 dBm e 10 dBm. Esta configuração foi utilizada em todos os estudos de caso descritos neste trabalho que utilizam inserção de interferência.

O rádio coletor e *tracker* ficaram muito próximos do analisador de RF, e distantes aproximadamente 2 metros do gerenciador de rede. Após o rádio coletor ter se agregado à rede WH, a seguinte sequência de testes foi realizada:

1) obtenção dos dados de medição de energia, durante 10 minutos, sem inserção de interferência na rede;
2) obtenção dos dados de medição de energia, durante 5 minutos, com inserção de interferência no canal 11 pelo analisador de RF;
3) obtenção dos dados de medição de energia, durante 10 minutos, sem inserção de interferência na rede.

A Figura 28 mostra os resultados obtidos com o teste realizado. Pode-se notar que nos períodos sem inserção de interferência, a energia medida no canal 11 ficou muito próxima a -100 dBm, indicando um canal praticamente livre de sinal. Os picos de energia que existiram nos períodos sem interferência são relativos às mensagens recebidas pelo rádio coletor no canal 11, ou seja, mensagens enviadas do gerenciador de rede para o rádio coletor. Dessa forma, verifica-se que, através da ferramenta proposta, podem-se identificar os momentos em que ocorreram recepções de mensagens em determinado canal.

Também se verifica que a ferramenta detectou corretamente a interferência inserida pelo analisador de RF, ou seja, a energia do canal 11 aumentou para aproximadamente -55 dBm durante o período de inserção de interferência.

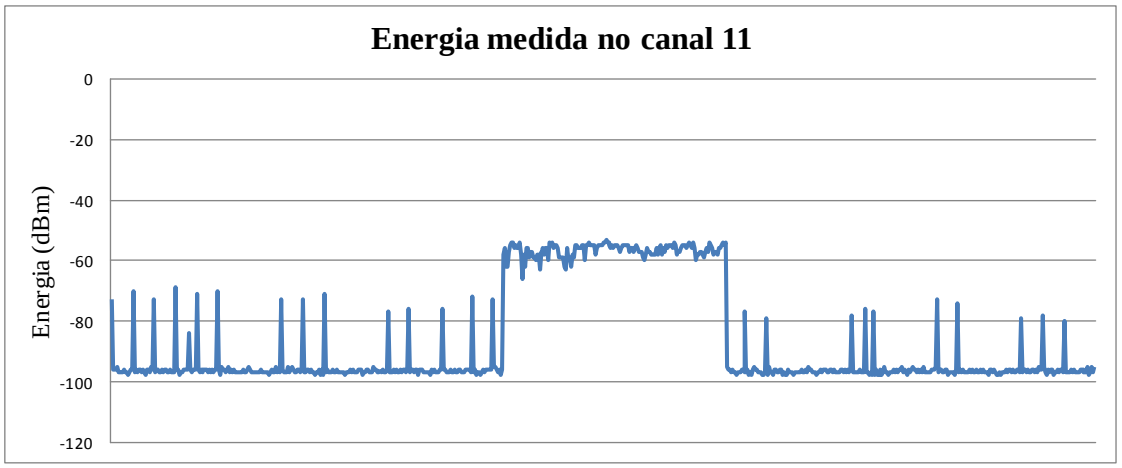


Figura 28 – Energia medida no canal 11 (com e sem interferência)

A fim de verificar que os outros canais não foram afetados pela inserção de interferência no canal 11, a Figura 29 mostra a energia medida nos 15 canais WH durante todo o tempo de teste (etapa sem inserção de interferência e com inserção de interferência). Pode-se notar que realmente apenas o canal 11 teve sua energia modificada em função da interferência inserida. Com isso, comprova-se que o *tracker* está, de fato, sendo sintonizado nas frequências corretas durante a medição de energia.

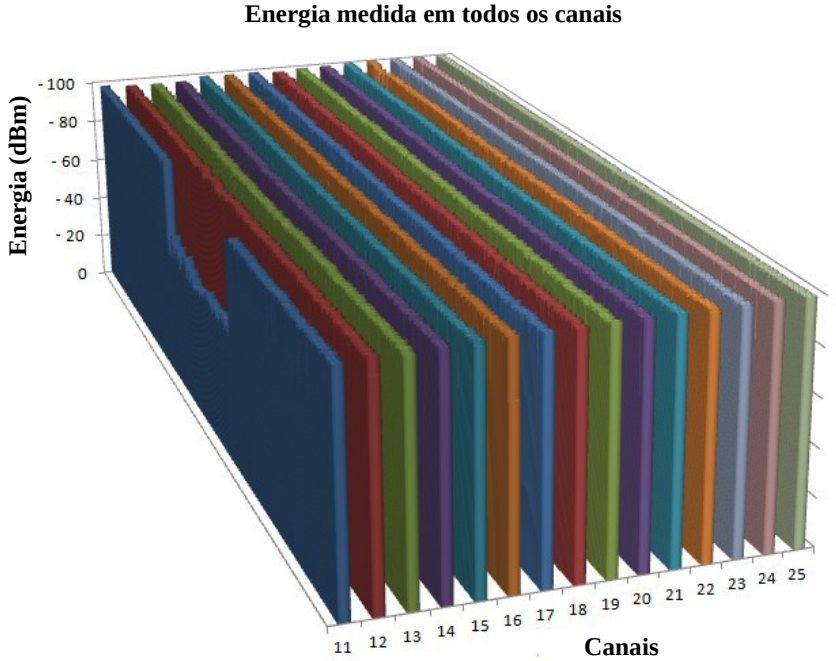


Figura 29 – Energia em todos os canais durante o período de interferência no canal 11

5.3 ESTUDO DE CASO 1: COMPARAÇÃO DO RSL DA MENSAGEM COM A ENERGIA MEDIDA

Após as validações experimentais já terem mostrado que a ferramenta proposta cumpre seus objetivos, ou seja, está capturando todos os dados necessários para realizar a decriptação das mensagens, está capturando as mensagens dos dispositivos ao seu alcance, e está detectando a variação de energia nos canais, será feita uma análise da comparação entre o RSL da mensagem recebida pelo rádio coletor e a energia medida no momento da recepção de tal mensagem. Além disso, serão obtidos dados quando a rede estiver sem interferência e com interferência.

Para obter os dados necessários, configurou-se o gerenciador de rede para utilizar apenas os canais 11, 12, 13, 14 e 15. Isso foi feito apenas para aumentar o número de mensagens recebidas em cada canal, ou seja, gerar mais amostras para os gráficos de análise dos resultados. Então, foram feitas medições de energia durante um intervalo de tempo em que a rede estava sem interferência e, logo após, foi inserida interferência no canal 11 utilizando o analisador de RF *FieldFox N9912A*. Por fim, a interferência foi removida novamente. A duração de cada uma destas etapas foi aproximadamente 10 minutos. Para realizar as medições, o rádio coletor já estava agregado à rede WH.

A Figura 30 mostra os valores de RSL das mensagens recebidas em cada canal e seus correspondentes valores de energia medida. A associação entre o valor de RSL e energia foi feita por meio do ASN, dessa forma, garante-se que a comparação está sendo feita de maneira correta, ou seja, a cada recepção de uma mensagem WH, tem-se um RSL associado e uma medição de energia no mesmo momento da recepção da mensagem.

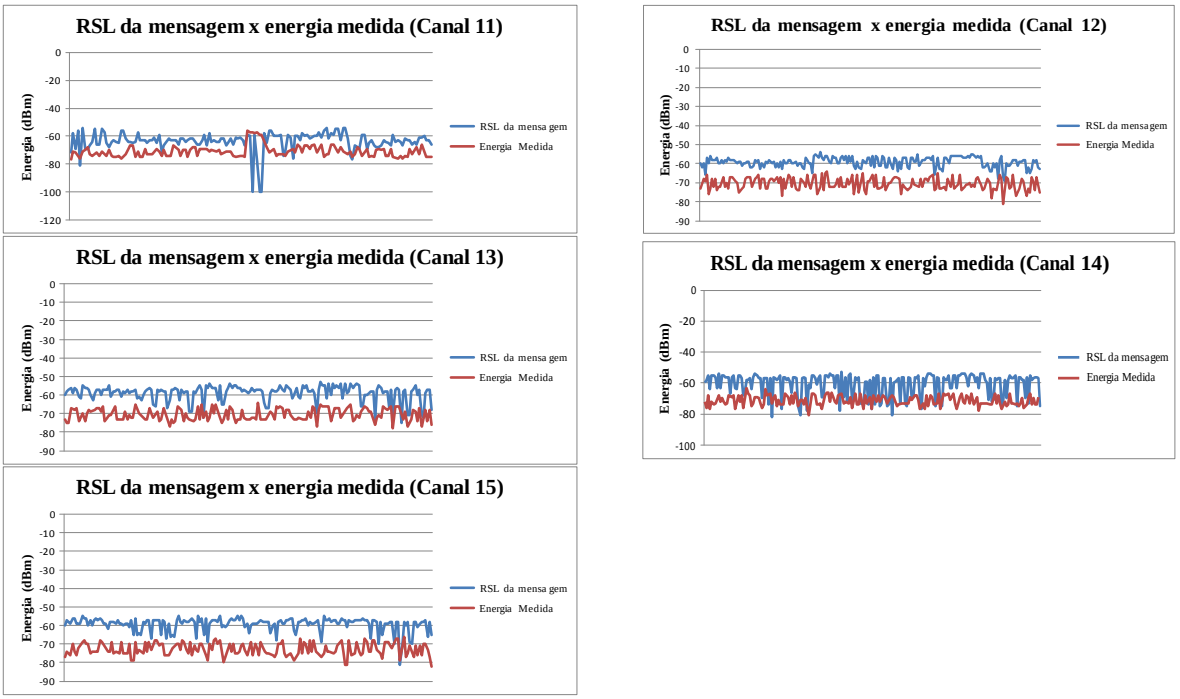


Figura 30 – RSL x energia nos canais 11 a 15

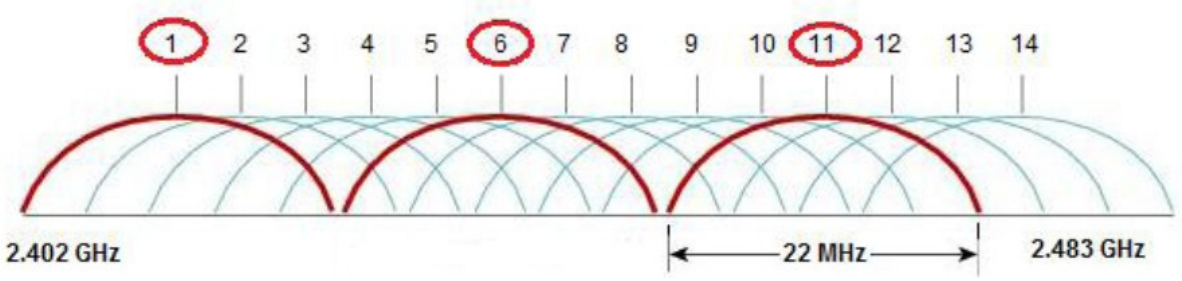
Pode-se perceber que o RSL das mensagens recebidas é superior à energia medida no canal. O cálculo do RSL considera apenas o sinal referente à mensagem WH, ou seja, não considera toda a energia presente em determinado canal. Para isso, realiza um tratamento no sinal, podendo-se citar filtragem e aplicação de ganhos.

Verifica-se que no momento da inserção de interferência no canal 11, o RSL de algumas mensagens reduziu para -100 dBm, que indica ausência de sinal, o que significa que, nestes casos, a mensagem WH deve ter sido perdida devido à interferência.

5.4 ESTUDO DE CASO 2: COEXISTÊNCIA DA REDE WH COM IEEE 802.11 (WI-FI)

A família IEEE 802.11 (Wi-Fi) consiste em uma série de padrões que operam nas faixas de frequências de 2,4 GHz, 3,6 GHz e 5 GHz, as quais são utilizadas para implementação das *Wireless Local Area Network* (WLAN). Os padrões mais populares dessa família são o IEEE 802.11b e IEEE 802.11g, os quais utilizam a faixa de frequência de 2,4 GHz, e definem um total de 14 canais, sendo que cada canal possui largura de banda de 22

MHz (WINTER, 2013). A Figura 31 ilustra os canais do padrão IEEE 802.11b/g. Pode-se ver que apenas três canais podem ser utilizados ao mesmo tempo, sem que haja sobreposição dos mesmos.



Fonte: (WINTER, 2013)

Figura 31 – Frequências utilizadas pelos padrões IEEE 802.11b/g

Então, uma vez que os canais do padrão IEEE 802.11b/g utilizam a mesma faixa de frequência do WH, ou seja, existe sobreposição de canais dos dois protocolos, um teste interessante é verificar o comportamento da ferramenta proposta neste trabalho durante a coexistência das duas redes.

Para realizar o teste, foi utilizado o cenário mostrado na Figura 32. A geração do tráfego Wi-Fi foi realizada utilizando a ferramenta *Distributed Internet Traffic Generator* (D-ITG) (BOTTA et al., 2012), de forma que são enviados pacotes da estação A para estação B. A estação A está conectada via cabo com o ponto de acesso, e a estação B está conectada via rede sem fio. Assim, os pacotes enviados pela estação A geram tráfego Wi-Fi a ser recebido pela estação B.

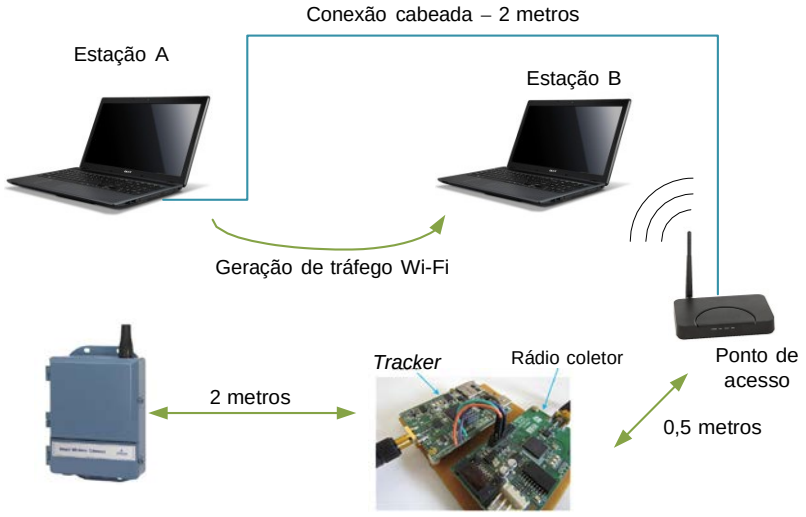
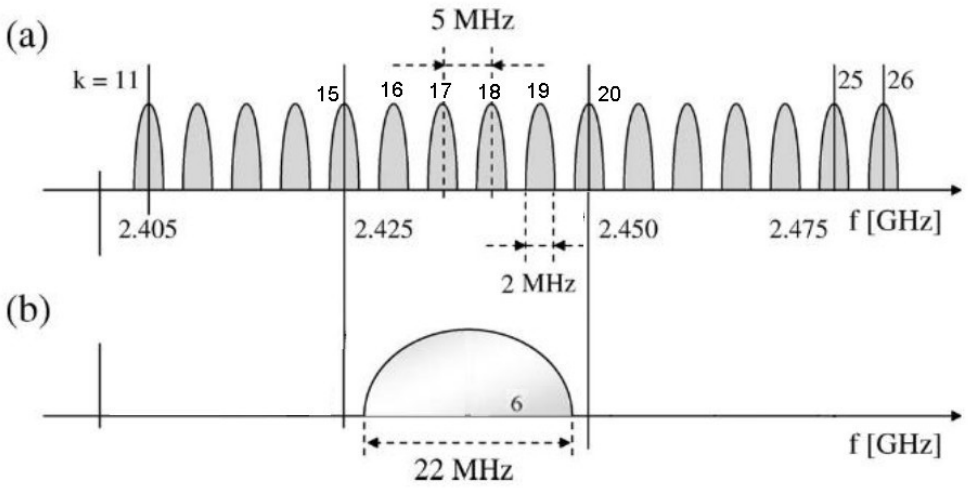


Figura 32 – Cenário de teste para análise de coexistência entre WH e IEEE 802.11g

O canal Wi-Fi utilizado foi o canal 6. Portanto, como pode ser visto na Figura 33, os canais 16 a 19 do WH provavelmente irão sofrer interferência da rede Wi-Fi.



Fonte: Adaptado de (WINTER, 2013)

Figura 33 – Sobreposição de canais. (a) Protocolo WH. (b) Protocolo IEEE 802.11.

Para realizar o teste, inicialmente foram coletados dados de medição de energia sem que a rede Wi-Fi estivesse sendo utilizada. Logo após, a ferramenta D-ITG foi inicializada para gerar tráfego Wi-Fi e, por fim, a rede manteve-se sem interferência novamente. A Figura

34 mostra o espectro de frequência obtido com o analisador de RF *FieldFox N9912A* durante a geração do tráfego Wi-Fi. Nota-se que, de fato, existe apenas geração de sinal no canal 6 do Wi-Fi.

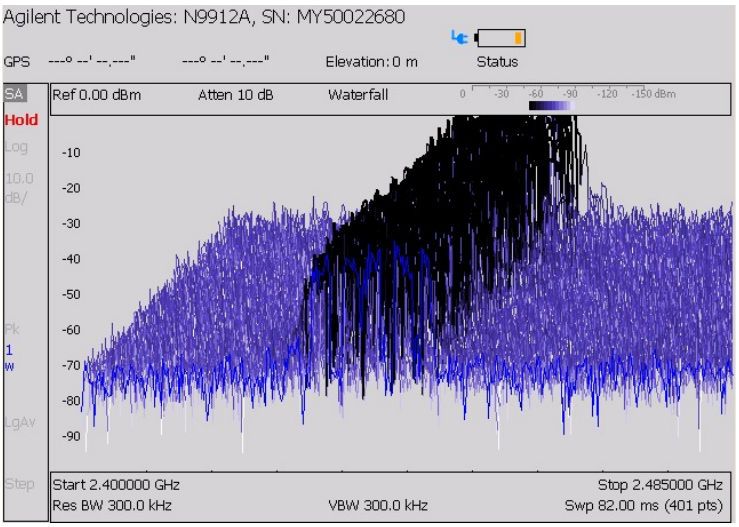


Figura 34 – Espectro de frequência durante a inserção de interferência Wi-Fi na rede WH

O resultado da medição de energia em todos os canais WH é mostrado na Figura 35. Nota-se que apenas os canais 16 a 19 tiveram sua energia afetada devido à inserção da interferência do Wi-Fi, o que era esperado, uma vez que o canal 6 do Wi-Fi deveria afetar somente estes canais, como foi mostrado na Figura 33.

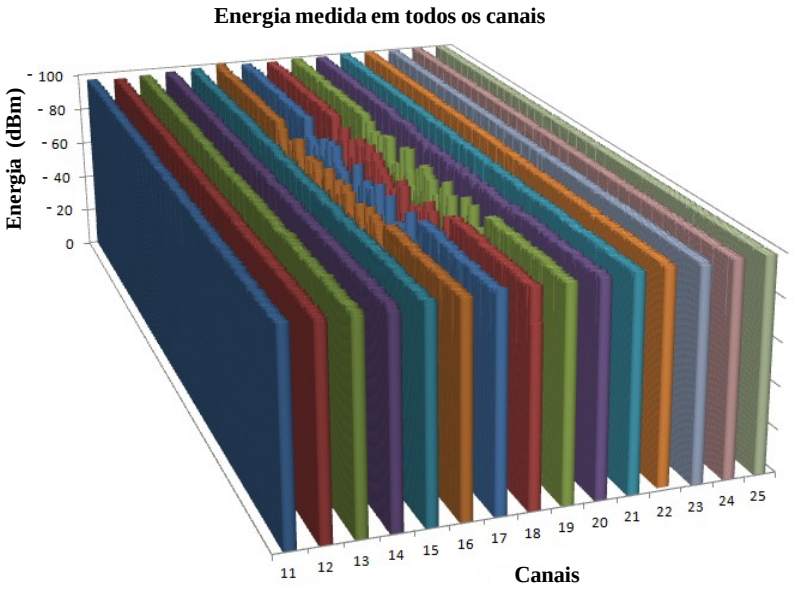


Figura 35 – Medição de energia em todos os canais durante o período de interferência Wi-Fi

Outra maneira de apresentar o resultado deste teste consiste em realizar a média de energia em cada um dos canais, durante todo o período do teste, de modo a verificar os canais que mais sofreram interferência do Wi-Fi. A Figura 36 ilustra o resultado. Pode-se perceber que a energia média dos canais ficou coerente com a largura de banda do canal 6 do Wi-Fi.

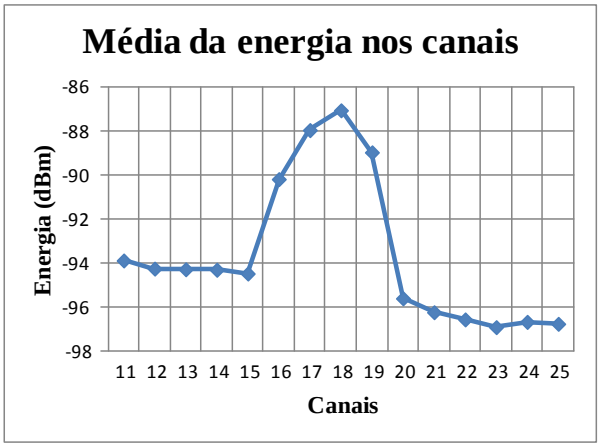


Figura 36 – Média da energia nos canais durante teste de coexistência do WH com Wi-Fi

6 CONCLUSÃO

Neste trabalho foi proposta uma ferramenta para monitoramento e análise de redes WH. A ferramenta atua como um *sniffer*, capturando todas as mensagens que trafegam pela rede e que estejam ao seu alcance. Tal ferramenta apresenta algumas vantagens em relação às ferramentas já propostas na literatura, como por exemplo: utilização de apenas um transceptor IEEE 802.15.4 para capturar dados nos 15 canais WH; armazenamento local das informações coletadas, por meio da utilização do cartão SD, fazendo com que não haja necessidade de conexão da ferramenta com uma estação de trabalho, permitindo seu uso em locais de difícil acesso; a unidade coletora de dados pode ser um nó roteador da rede; e, por fim, a ferramenta realiza constantemente a medição de energia nos canais que estão sendo utilizados a cada instante de tempo (*time slot*), permitindo que sejam feitas análises a respeito de interferências na rede.

Além da ferramenta de coleta dos dados em campo, também foi desenvolvida uma aplicação para análise *offline* dos dados coletados (foram realizadas modificações em um ambiente já desenvolvido anteriormente). Nessa ferramenta são apresentadas ao usuário algumas análises e informações estatísticas a respeito da rede. As mensagens são classificadas de acordo com o seu tipo e é feita uma contabilização do número de mensagens de cada tipo que ocorreram em cada um dos canais. Também é mostrado o número de mensagens que tiveram perda de ACK e em quais canais isso ocorreu. Outra informação interessante diz respeito à sincronização temporal entre os dispositivos, de forma que a ferramenta apresenta o ajuste temporal entre cada par de dispositivos que trocaram mensagens durante o período de coleta dos dados.

As validações experimentais apresentadas mostraram que a ferramenta desenvolvida alcançou os objetivos propostos, ou seja, a captura das mensagens dos dispositivos vizinhos que estejam ao seu alcance (*sniffer*) e a medição de energia nos canais. Além disso, a

ferramenta de análise *offline* dos dados está realizando corretamente a interpretação dos dados e apresentando as análises e estatísticas de forma clara.

Os estudos de caso foram voltados para a capacidade da ferramenta realizar a detecção de energia nos canais. Esta é uma funcionalidade muito importante da ferramenta proposta, pois fornece informações a respeito de interferências na rede. Uma vez que o espectro de 2,4 GHz é bastante utilizado por outros protocolos, como por exemplo, Wi-Fi, é importante que se conheça o comportamento da rede WH durante a coexistência do WH com outros protocolos.

Um dos estudos de caso permitiu a comprovação de que a ferramenta está realizando a medição de energia corretamente, uma vez que detectou a interferência inserida. Além disso, apenas analisando visualmente a curva de energia gerada para cada um dos canais, pode-se perceber a ocupação do canal, ou seja, os momentos em que ocorreram comunicações nos canais.

O segundo estudo de caso apresentou o comportamento da ferramenta quando ocorre a coexistência do WH com o Wi-Fi. O tráfego Wi-Fi foi gerado no canal 6, que é sobreposto aos canais WH 16 a 19. Pelos resultados, viu-se que realmente apenas estes canais foram afetados pela interferência gerada pelo Wi-Fi, e os demais canais não sofreram impacto.

Os trabalhos futuros sugeridos dizem respeito à aplicação da ferramenta proposta, principalmente no que diz respeito à detecção de energia na rede. Uma vez que se tem uma ferramenta que detecta a energia no momento em que uma mensagem está sendo recebida, algumas análises importantes podem ser realizadas, como por exemplo, verificar se houve perda de ACK devido ao nível de energia do canal, ou seja, se as mensagens foram corrompidas devido a interferências na rede. Também se pode determinar até que ponto uma interferência não impacta na integridade das mensagens.

Outra questão importante refere-se ao local onde é realizada a medição de energia. A ferramenta proposta realiza a detecção de energia no ponto físico em que se encontra. No entanto, como a ferramenta atua como um *sniffer*, capturando mensagens de outros dispositivos, acaba realizando também a detecção de energia nos canais envolvidos nestas outras comunicações. Porém, estes outros dispositivos que estão se comunicando podem estar fisicamente afastados do ponto de coleta dos dados, que é o mesmo ponto da medição de energia. Com isso, é preciso fazer um estudo sobre como interpretar a energia medida pela ferramenta durante as comunicações que envolvem dispositivos distantes do ponto de coleta.

REFERÊNCIAS

BOTTA, A.; DAIANOTTI, A.; PESCAPÉ, A. A tool for the generation of realistic network workload for emerging networking scenarios. **Journal Computer Networks**, [S.l.], v. 56, p. 3531-3547, 2012.

CHEN, D., NIXON, M., MOK, A.K. **WirelessHART:** Real-Time Mesh Network for Industrial Automation**.** New York: Springer, 2010. 276 p. ISBN 978-1-4419-6046-7.

DEPARI, A. et. al. Design and Performance Evaluation of a distributed *Wireless*HART Sniffer Based on IEEE1588**.** In: INTERNATIONAL IEEE SYMPOSIUM ON PRECISION CLOCK SYNCHRONIZATION FOR MEASUREMENT, CONTROL AND COMMUNICATION (ISPCS), 2009, Brescia. **Proceedings . . .** New York: IEEE, 2009. p. 1-6.

FREESCALE SEMICONDUCTOR. **MC1322x Software Driver:** Reference Manual. 2011a. 300 p. Disponível em: <cache.freescale.com files rf if doc ref manual 22 DR RR .pdf>. Acesso em: 18 abr. 2013.

FREESCALE SEMICONDUCTOR. **MC1322x Simple Media Access Controller:** Reference Manual. 2011b. 48 p. Disponível em: <cache.freescale.com/files/rf_if/doc/ref_manual/ 22 S ACR .pdf>. Acesso em: 23 abr. 2013.

FREESCALE SEMICONDUCTOR. **MC1322x:** Advanced ZigBee Compliant SoC Platform for the 2.4 GHz IEEE 802.15.4 Standard**:** Reference Manual. 2012. 532 p. Disponível em: <www.freescale.com/files/rf_if/doc/ref_manual C1322 R .pdf>. Acesso em: 1 nov. 2012.

HAN, S. et. al. Wi-HTest: Compliance Test Suite for Diagnosing Devices in Real-Time *Wireless*HART Network**.** In: IEEE REAL-TIME AND EMBEDDED TECHNOLOGY AND APPLICATIONS SYMPOSIUM (RTAS), 2009, São Francisco. **Proceedings . . .** New York: IEEE, 2009. p 327-336.

HART COMMUNICATION FOUNDATION (HCF). **HCF_SPEC-065:** 2.4GHz DSSS O-QPSK Physical Layer Specification. Austin, 2007, 20 p.

HART COMMUNICATION FOUNDATION (HCF). **HCF_SPEC-075:** TDMA Data Link Layer Specification. Austin, 2008, 76 p.

HART COMMUNICATION FOUNDATION (HCF). **HCF_SPEC-085:** Network Management Specification. Austin, 2009, 98 p.

HUITRÓN G.D.; GALLARDO J.H. **SMAC:** Wireless connectivity made easy**.** Disponível em: <www.freescale.com/webapp/sps/site/overview.jsp?code=784_LPBBSMAC>. Acesso em 23 abr. 2013.

INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS (IEEE). **IEEE Standard for Local and Metropolitan Area Networks - Part 15.4:** Low-Rate Wireless Personal Area Networks (LR-WPANs). New York, 2011, 314p.

JONSSON, M.; KUNERT, K. Towards Reliable Wireless Industrial Communication with Real-Time Guarantees. **IEEE Transactions On Industrial Informatics**, New York, v.5, n.4, p.429-442, 2009.

KRÄTZIG, M. et. al. 16-Channel-Analyser for Parallel IEEE 802.15.4 Monitoring. In: IEEE CONFERENCE ON EMERGING TECHNOLOGIES & FACTORY AUTOMATION (ETFA), 2009, Mallorca. **Proceedings . . .** New York: IEEE, 2009. p. 1-4.

KUNZEL, G. **Ambiente para Avaliação de Estratégias de Roteamento para Redes WirelessHART**. 2012. 95 p. Dissertação (Mestrado em Engenharia Elétrica) – Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2012.

LIMA, C. et. al. Porta de Manutenção para Comissionamento e Análise Local de Redes WirelessHART. In: CONGRESSO BRASILEIRO DE AUTOMÁTICA (CBA), 2012, Campina Grande. **Anais . . .** Campinas: SBA, 2012. p. 4923-4929.

LORENÇATO, A. et. al. *Wireless*HART Field Device with Integrated Network Analyser. In: SYMPOSIUM ON COMPUTING AND AUTOMATION FOR OFFSHORE SHIPBUILDING (NAVCOMP), 2013, Rio Grande. **Proceedings . . .** Rio Grande: IEEE, 2013. p. 29-33.

LORENÇATO, A. **Analisador de redes WirelessHART.** 2013. 106 p. Dissertação (Mestrado em Engenharia Elétrica) – Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2013.

MULLER, I. et al. Development of a WirelessHART Compatible Field Device. In: IEEE INSTRUMENTATION AND MEASUREMENT TECHNOLOGY CONFERENCE (I2MTC), 2010, Austin. **Proceedings . . .** New York: IEEE, 2010. p. 1430-1434.

MULLER, I. et al. Namimote: a low-cost sensor node for wireless sensor networks. In: INTERNATIONAL CONFERENCE ON NEXT GENERATION WIRED/WIRELESS ADVANCED NETWORKS AND SYSTEMS (NEW2AN), 2012, São Petersburgo. **Proceedings. . .** New York: Springer, 2012. p.391–400. (Lecture Notes in Computer Science, v.7469).

MULLER, I.; PEREIRA C.E.; NETTO, J. *Wireless*HART Field Devices**. IEEE Instrumentation and Measurement Magazine**, [S. l.], v.14, p. 20-25, Dec. 2011.

MULLER, I. **Gerenciamento Descentralizado de Redes Sem Fio Industriais Segundo o Padrão *Wireless*HART.** 2012. 105p. Tese (Doutorado em Engenharia Elétrica) – Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2012.

SD GROUP, SD CARD ASSOCIATION. **SD Specifications Part 1:** Physical Layer Simplified Specification Version 4.10**.** 2013. 202 p. Disponível em: <www.sdcard.org/downloads/pls/simplified_specs/part1_410.pdf>. Acesso em: 1 ago 2013.

WINTER, J. et. al. *Wireless*HART Routing Analysis Software. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SISTEMAS COMPUTACIONAIS (SBESC), 2011, Florianópolis. **Anais . . .** [S.l.: s.n.], 2011. p. 206-211.

WINTER, J. **Análise de coexistência em redes *Wireless*HART.** 2013. 103p. Dissertação (Mestrado em Engenharia Elétrica) – Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2013.