UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

JEAN MICHEL WINTER

PROJETO DE DIPLOMAÇÃO

SOFTWARE DE ANÁLISE DE ROTEAMENTO DE DISPOSITIVOS WIRELESSHART

Porto Alegre
(2010)

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

SOFTWARE DE ANÁLISE DE ROTEAMENTO DE DISPOSITIVOS WIRELESSHART

Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para Graduação em Engenharia Elétrica.

ORIENTADOR: Prof. Dr. Carlos Eduardo Pereira

Porto Alegre
(2010)

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

JEAN MICHEL WINTER

SOFTWARE DE ANÁLISE DE ROTEAMENTO DE DISPOSITIVOS WIRELESSHART

Este projeto foi julgado adequado para fazer jus aos créditos da Disciplina de "Projeto de Diplomação", do Departamento de Engenharia Elétrica e aprovado em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____

Prof. Dr. Carlos Eduardo Pereira, UFRGS

Doutor pela Technische Universitat Stuttgart – Stuttgart, Alemanha

Banca Examinadora:

Prof. Dr. Carlos Eduardo Pereira, UFRGS

Doutor pela (Technische Universitat Stuttgart – Stuttgart, Alemanha)

Msc. Ivan Muller, UFRGS

Mestre em Engenharia Elétrica (UFRGS - Porto Alegre, Brasil)

Prof. Dr. Marcelo Götz, UFRGS

Doutor pela (Universität Paderborn – Paderborn, Alemanha)

Porto Alegre, Dezembro de 2010.

AGRADECIMENTOS

Agradeço a todos que de alguma forma contribuíram para a conclusão deste trabalho em especial aos meus pais Líria e Renato e meus irmãos Clóvis e Daniel pelo incentivo e apoio.

Aos amigos e colegas do curso de Engenharia Elétrica Aramis Tisott , Caciano Machado, Caetano Lazzari, Felipe Faccin, Fabrizio Duarte Tissot, Henrique Girardi Hall e Rodolfo de Freitas Valle Dresch.

Aos colegas e amigos do Laboratório de Automação, Sistemas de Controle e Robótica em especial ao Professor Carlos Eduardo Pereira e Reiner Franthesco Perozzo.

Ao Professor João Cesar Netto, Ivan Muller e Alexandre Lorencato que também contribuíram para o desenvolvimento deste projeto.

RESUMO

Atualmente os dispositivos utilizados em processos industriais como, por exemplo, sensores e atuadores têm sido controlados por meio de cabeamento estruturado aumentando os custos e às vezes inviabilizando o controle ou monitoramento do processo em determinado local da planta. Este trabalho tem como objetivo principal realizar o estudo de um protocolo de comunicação sem fio para aplicação em processos industriais através do desenvolvimento de uma ferramenta de software para a análise das principais características e comportamento do protocolo WirelessHART.

Palavras-chaves: WirelessHART. Protocolos de Comunicação Sem Fio. Roteamento e Automação e Controle.

ABSTRACT

Currently the devices involved in industrial processes like sensors and actuators have been controlled through wired structures rising costs and sometimes making it impossible to control or monitor the process at a particular location of the industrial plant. The main goal of this document is conducting a study of a communication protocol for wireless application in industrial processes by developing a software tool that makes possible the analysis of the key characteristics and behavior of this protocol.

Keywords: WirelessHART. Wireless Communication Protocol. Routing. Automation and Control.

SUMÁRIO

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS

ACK: *Acknowledge*

ASN: *Absolute Slot Number*

CRC: *Cyclic redundancy check*

DD: *Device Description*

DSSS: *Direct Sequence Spread Spectrum*

FSK: *Frequency Shift Key*

HCF: *Hart Communication Foundation*

IEC: *International Electrotechnical Commission*

ISM: *Industrial, Scientific and Medical*

LLC: *Logical Link Control*

MAC: *Medium Access Control*

MIC: *Message Integrity Code*

NPDU: *Network Protocol Data Unit*

PDU: *Protocol Data Unit*

UFRGS: Universidade Federal do Rio Grande do Sul

1 INTRODUÇÃO

Os avanços na área da eletrônica permitiram o crescimento tecnológico dos equipamentos de automação para processos industriais. Equipamentos mais robustos, confiáveis e integrados com processadores de maior poder computacional dispõem de maior fluxo de informações e recursos tornando cada vez mais relevante a comunicação entre os diversos dispositivos que compõem o cenário da indústria.

O controle de processos industriais tem passado por gerações de avanço em tecnologias de comunicação desde a comunicação pneumática até a eletrônica. Desde a década de 80 existem grandes esforços por parte da indústria em projetar e especificar protocolos de comunicação que atendam os diversos requisitos de processo entre eles confiabilidade, segurança e durabilidade. Melhorias na tecnologia de sensores e diagnósticos levaram ao surgimento de uma grande variedade de dispositivos inteligentes, dispositivos dotados de microprocessadores os quais permitem aplicações mais complexas fornecendo uma visão maior do processo e melhoria no desempenho operacional da planta. Atualmente os fabricantes de atuadores e sensores não oferecem uma tecnologia de comunicação sem fio que possibilite a automação destes dispositivos em locais onde não seja possível o lançamento de dutos e passagem de cabos de comunicação. Os ganhos efetivos de custos e acesso confiável a cada canto da planta, não apenas aos pontos críticos, são os incômodos das redes baseadas em cabeamento.

Algumas organizações industriais tem promovido o uso de tecnologias sem fio na indústria, várias tentativas de lançar um padrão para uso industrial entre elas (Bluetooth, WiFi, Zigbee, WINA) os quais não obtiveram uma aceitação definitiva por parte da indústria principalmente pela robustez do padrão. O ambiente industrial no qual as redes são instaladas é extremamente hostil, uma vez que ruídos eletromagnéticos de grande intensidade podem

estar presentes (por exemplo, no acionamento de motores elétricos, em função das altas correntes envolvidas, radiações eletromagnéticas são geradas, podendo induzir ruídos nos equipamentos eletrônicos nas proximidades). Além disso, ambientes industriais também costumam apresentar temperaturas e umidades elevadas, dois aspectos prejudiciais aos componentes utilizados em sistemas computacionais e de comunicação. Desta forma, os equipamentos designados para uso em redes industriais são em geral especialmente construídos para trabalhar nestas condições adversas e os protocolos de comunicação adotados também devem considerar aspectos de segurança e disponibilidade do sistema desenvolvido.

O presente trabalho apresenta um estudo sobre o protocolo de comunicação WirelessHart e o desenvolvimento de uma ferramenta de software que viabilize a análise das principais características e comportamento deste protocolo. Esta ferramenta visa possibilitar uma melhor análise do comportamento da topologia de rede entre os dispositivos WirelessHart permitindo possíveis melhorias na robustez deste protocolo e possibilitar ainda mais o processo de aceitação deste padrão para uso industrial atendendo de modo confiável os requisitos da indústria.

2 COMUNICAÇÃO INDUSTRIAL

Com a crescente tendência na área de automação industrial do uso de arquiteturas computacionais distribuídas, onde diferentes dispositivos podem estabelecer comunicação com outros de forma a integrar diversos equipamentos e inclusive de fabricantes distintos (protocolo aberto) causou o crescimento e aprimoramento nos usos das redes de automação. A utilização das redes permite a comunicação rápida e confiável entre equipamentos e o uso de mecanismos padronizados, que são hoje em dia, fatores indispensáveis no conceito de produtividade industrial.

Entre os protocolos utilizados na indústria podemos citar *fieldbus*, modbus, ASI, Can, Device Net, Hart entre tantos outros os quais tem seu uso especificado de acordo com os requisitos do sistema (custo, equipamentos utilizados, tempo de resposta, confiabilidade, etc.). Estas tecnologias começaram de um modo bastante simples, desde quando a comunicação utilizava o padrão serial ou paralelo até padrões de comunicação sem fios (WirelessHart, por exemplo) sendo os protocolos desenvolvidos para cada aplicação dependendo da rede utilizada e para cada caso de transmissão logo que as redes diferem em função das condições físicas e mecânicas.

Entre as décadas de 60 e 70 ainda não havia a transmissão de dados ou informações representadas por sinais analógicos de 0-10V e 4-20 mA, somente na década de 80 surgiu a transmissão digital de dados e o uso de microprocessadores implementando protocolos de comunicação[1]. Nos próximos capítulos será detalhado um pouco mais sobre as características dos protocolos Hart e WirelessHart.

3 HART COMMUNICATION FOUNDATION

Fundada em 1993, a HART Communication Foundation (HCF), uma organização internacional, sem fins lucrativos, é a proprietária e autoridade central da tecnologia sobre o protocolo HART [2]. A Fundação gerencia e controla as normas de protocolo, incluindo desenvolvimento de novas tecnologias e melhorias.

O foco principal de todas as atividades da fundação é promover a aplicação da tecnologia HART, reforçar a sua posição no mercado global e ajudar os usuários a maximizar o valor dos seus investimentos em instrumentação inteligente [2].

3.1 CRIAÇÃO DA FUNDAÇÃO

Em 1990 ocorreu a primeira reunião oficial do que então era chamado HART Users Group. Esta etapa marcou o inicio do HART como uma tecnologia de comunicação aberta.

Representantes de 26 empresas reuniram-se no primeiro encontro onde foi discutido sobre a aplicação da tecnologia HART com foco nas camadas de aplicação, enlace e física. Foi discutido a organização de um grupo que faria a gestão dos padrões de protocolo e prestaria apoio em nível mundial para a tecnologia.

Entre as empresas participantes estavam:

- Exxon Química;
- ABB Kent-Taylor;
- Rosemount Sherex;
- Siemens;
- SMAR;
- Proctor & Gamble;
- Yokogawa;
- Hitachi.

Ainda em 1990 as empresas associadas nomearam quatro grupos de trabalho: Definição, Testes de Conformidade, Interface Homem-Máquina e Interoperabilidade. Na seqüência ainda foram adicionados mais dois grupos de trabalho: Saída e PID. Finalmente em 1993 o Grupo de Usuários HART acordou na criação de uma organização independente,

sem fins lucrativos para gerenciar e suportar o protocolo HART e então surgiu a HART Communication Foundation.

3.2 TECNOLOGIA E MELHORAMENTOS

Desde a criação da fundação Hart existiram esforços juntamente com as empresas associadas e com os utilizadores na indústria para desenvolver melhorias para o protocolo HART de modo a atender as necessidades da indústria de comunicação de dispositivos e criar ferramentas de desenvolvimento adicional aos produtos.

A fundação apresentou o servidor HART, em 1999, uma aplicação de software OPC (*OLE for Process Control*) que fornece uma portal fácil de usar para acessar em tempo real processos e informações de diagnóstico disponíveis na instrumentação HART de campo.

Em 2001, o primeiro grande reforço do protocolo foi lançado, o HART 6. Incluindo o suporte para as *tags* longas, comandos adicionais além de comunicação digital mais rápida.

Desde 1990 o dispositivo HART de descrição de linguagem, *Device Description Language* (DDL) tem sido um elemento chave da tecnologia HART. Em 2004, o DDL foi aprovado pela IEC como um padrão internacional [2]. Hoje ele continua a ser a linguagem de comunicação mais importante e amplamente utilizada[2].

Em 2004, lançou o HART DD-IDE (Device Description Integrated Development Environment) com o *Smart Device Configurator* (SDC-625), um conjunto integrado de ferramentas de apoio a teste, desenvolvimento e manutenção de DDs (*Device Description*) para dispositivos HART[2].

Trabalhando em colaboração com outras organizações internacionais de *fieldbus* e fornecedores de sistemas de automação e dispositivos, a fundação completou melhorias para a DDL em 2005, tornando a configuração do dispositivo um padrão.

Em 2006, a Fundação criou um Grupo de Usuários HART e atualmente está trabalhando em cooperação com outros grupos da indústria para encontrar formas de integrar a tecnologia HART para o benefício de toda a indústria [2].

Em 2007, o último passo na evolução do protocolo HART a norma HART 7, levava a tecnologia para o mundo da comunicação sem fio. A norma HART 7 inclui o novo padrão WirelessHART e aborda as necessidades críticas da indústria de processo para uma tecnologia segura, confiável e simples que fornece uma solução de bom senso e de esforços da indústria para uma solução de redes de comunicação sem fio [2]. Em 2010, a tecnologia WirelessHART foi reconhecido pela IEC (*International Electrotechnical Commission*) como primeiro padrão completo internacional de comunicação sem fio para processos na indústria[3].

3.3 PROGRESSO DA TECNOLOGIA

1994 Programa de Registro e Biblioteca DD DD;

1995 Especificações Teste HART, Procedimentos e do Programa de Registro de Dispositivos HART OPC Server 1999;

2001 Diagnósticos de reforços para sistema de integração com HART 6;

2003 Ambiente de Desenvolvimento Integrado DD;

2004 Smart Device Tecnologia de Software Configurador-SDC-625;

2005 Língua reforçada DD;

2006 Protocolo HART Normas Aprovadas pelo IEC (IEC 61158);

2007 HART 7 com WirelessHART;

2008 Receptor de Teste HART Wi-Analys e Analisador de Rede Wi-Analys

OPC Server 2009 para atualizar HART 7;

2010 WirelessHART IEC aprovado (IEC 62591Ed. 1.0)

4 PADRÃO HART

O protocolo de comunicação Hart é mundialmente conhecido como um padrão da indústria para comunicação de instrumentos de campo inteligente [1]. Uma das principais características deste padrão é a sobreposição do sinal de comunicação digital aos sinais analógicos o qual permite o uso de instrumentos inteligentes em cima dos tradicionais cabos 4-20mA. Dessa maneira é possível estender o uso do protocolo Hart, virtualmente, a todos os sistemas de controle de processos de plantas, viabilizando a medição de processos de maneira mais eficaz e interativa do que somente a instrumentação analógica. Através do uso deste protocolo são possíveis: diagnósticos sobre o dispositivo com o equipamento ao qual está anexado, monitoramento de processos, etc. Denominam-se *smart* os dispositivos capazes de executarem esta comunicação híbrida.

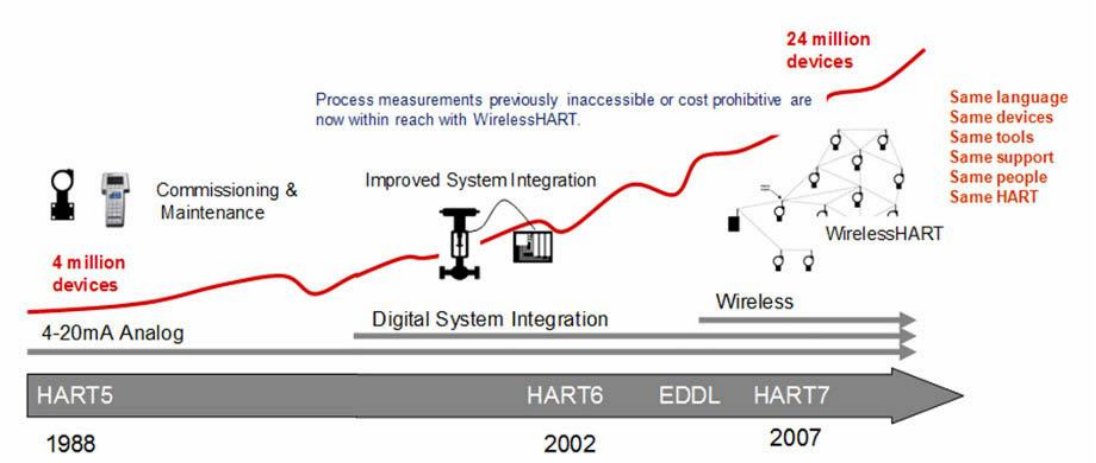

Figura 1. Evolução do Padrão HART.

4.1 PROTOCOLO DE 7 CAMADAS MODELO OSI

A Figura 2 apresenta o padrão HART comparado com as camadas do modelo do protocolo OSI (***O**pen **S**ystems **I**nterconnection*). Este modelo é dividido em camadas

hierárquicas sendo que cada camada usa as funções da própria camada ou da camada anterior, para esconder a complexidade e transparecer as operações para o usuário, seja ele um programa ou uma outra camada.

O protocolo WirelessHART compartilha a mesma camada de aplicações que o padrão HART utiliza, no entanto possui sua própria camada física, de enlace e de rede.

Camada OSI	Função	HART	
7 Aplicação	Fornece aos usuários as aplicações da rede.	Comandos orientados. Tipos de dados pré-definidos. Procedimentos de aplicações.	
6 Apresentação	Converte dados de aplicação entre a rede e a máquina local		
5 Sessão	Conecta os serviços de gerenciamento com as aplicações.		
4 Transporte	Transfere mensagens de forma transparente e independente na rede.	Transferência de conjunto de dados. Segmentação de dados. Negociação da segmentação de dados.	
3 Rede	Roteamento dos pacotes de ponta a ponta. Endereçamento da rede.		Otimização de energia, redundância de caminhos. Auto organização da rede.
2 Enlace	Estabelece pacote da estrutura de dados. Detecção de erros.	Conexão Elétrica/mecânica, transmissão de bits.	Segurança e confiabilidade. Tempo de sincronização. TDMA/CSMA.
1 Física	Conexão Elétrica / mecânica e transmissão.	Sinal analógico e digital simultaneamente.	Wireless 2.4 GHz, baseado em rádios 802.15.4, 10dBm.
		Com fio FSK/PSK e RS485	Sem fio 2.4GHz

Figura 2. Modelo OSI de 7 Camadas

4.2 PROTOCOLO HART

O protocolo Hart implementa as camadas 1,2, 4 e 7 do modelo OSI (Open Systems Interconnection) modelo de protocolo de 7 camadas [1], (ver Figura 2). A mensagem que contém a informação passa através destas camadas em um dispositivo, através do cabo, em seguida nas camadas correspondentes no outro dispositivo (Figura 3). As camadas 3 a 6 do

modelo OSI não são necessárias sendo muito reduzidas em ambientes restritos de uma rede local com operação entre dispositivos mestre-escravo com base em transações únicas e sem respostas automáticas para mensagens perdidas ou corrompidas. Essa simplificação leva a uma simplicidade notável em relação a outros protocolos *fieldbus* [1].

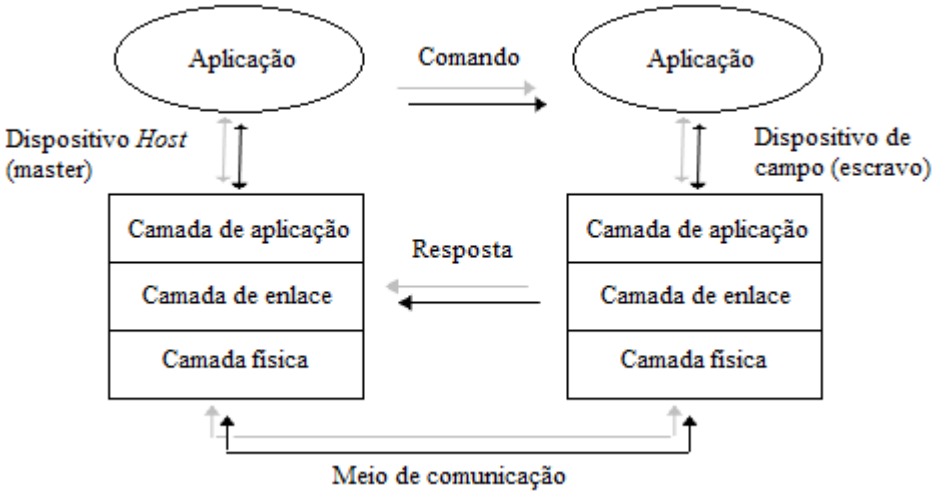

Figura 3. Protocolo HART e Modelo OSI.

O protocolo *wired* HART é baseado no padrão Bell 202, onde sinal é modulado em FSK (*Frequency Shift Keying*) sendo sobreposto ao sinal analógico de 4-20 mA. Para transmitir 1 e 0 são utilizados um sinal de 1 mA pico a pico nas freqüência de 1200 Hz e de 2400 Hz respectivamente [3]. A onda senoidal sobreposta no sinal DC (Direct Current) tem um valor médio de zero, sendo que nenhum componente DC é adicionado ao sinal de 4 a 20mA. Normalmente alguns instrumentos como um filtro passa-baixas adequado no sinal analógico pode ser utilizado para remover o sinal de comunicação e garantir nenhuma interferência do FSK. Este protocolo permite comunicação bidirecional possibilitando a transmissão e recepção de informações adicionais, além da normal, que é a variável de processo em instrumentos de campo inteligentes.

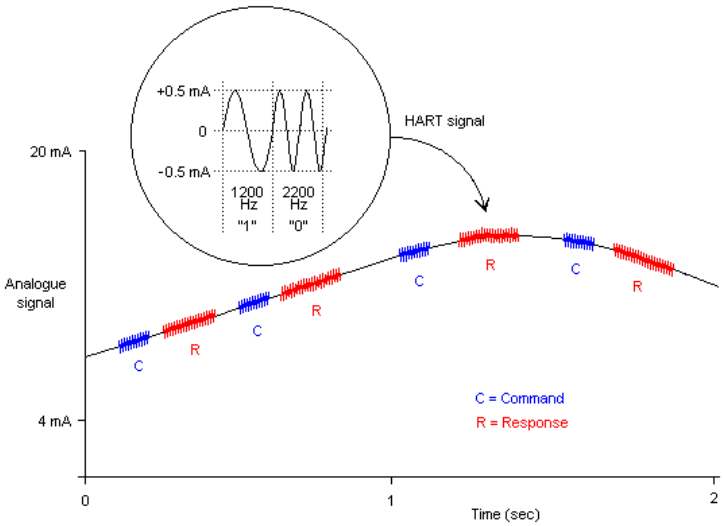

Figura 4. Sinal HART modulado em FSK (*Frequency Shift Keying*).

5 WIRELESHART

O padrão WirelessHart é o primeiro padrão aberto de comunicação para medidas e controle em processos industriais e faz parte da versão Hart 7. Usando uma rede sem fio tipo malha entre os dispositivos de campo, o protocolo WirelessHART é empregado em sensores e atuadores. O WirelessHART nasceu da necessidade de manter um padrão já consolidado na indústria (Hart) de forma a proteger os investimentos dos usuários em sua base instalada e da necessidade de usar tecnologia wireless para reduzir custos de medição, acesso de informações de diagnóstico avançado e propiciando um melhor acompanhamento do equipamento.

A tecnologia WirelessHart apresenta uma rede segura e opera na banda de rádio de 2,4 GHz ISM (*Industrial, Scientific and Medical*). O padrão utiliza a norma IEEE 802.15.4 (camada física) com seqüência direta de espalhamento do espectro, *Direct Sequence Spread Spectrum* (DSSS)[4]. A rede WirelessHart suporta uma ampla variedade de dispositivos de diversos fabricantes (ver Figura 5), incluindo:

- Dispositivos de campo como dispositivos básicos realizando funções de sensoriamento ou atuação;
- Dispositivos roteadores de campo, utilizados principalmente como roteadores das mensagens entre os dispositivo;
- Adaptador de dispositivos de campo, que liga os dispositivos HART com fio dentro da rede;
- Dispositivos portáteis para usuários móveis;
- Pontos de acesso que ligam os dispositivos de campo com um gateway;
- Gerente de rede simples (podem ser redundantes), que pode estar integrado no gateway ou aplicado em um *host.*

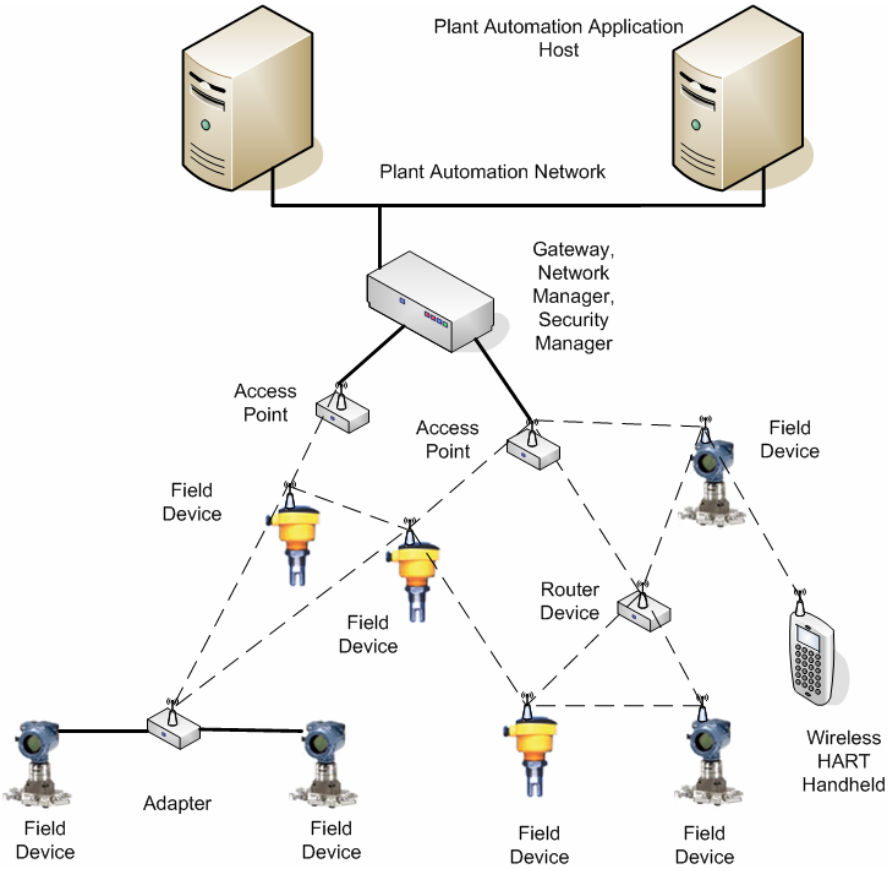

Figura 5 Exemplo de rede WirelessHART

O protocolo de comunicação WirelessHART é precisamente programado utilizando uma abordagem denominada Acesso Múltiplo por Divisão no Tempo (TDMA). A grande maioria das comunicações são dirigidas ao longo de rotas em grafos. O agendamento é realizado por um gerenciador de rede centralizado que utiliza as informações sobre a rede em combinação com os requisitos de comunicação fornecidos pelos dispositivos e pelas aplicações. A programação é dividida em *slots* (espaços no tempo) e transferida a partir do gerenciador de rede para dispositivos individuais. O gerente de rede, de forma contínua, adapta todo o roteamento e programa mudanças na topologia da rede e a demanda de comunicação[3]. É possível verificar a relação dos componentes dentro da comunicação WirelessHart na Figura 6.

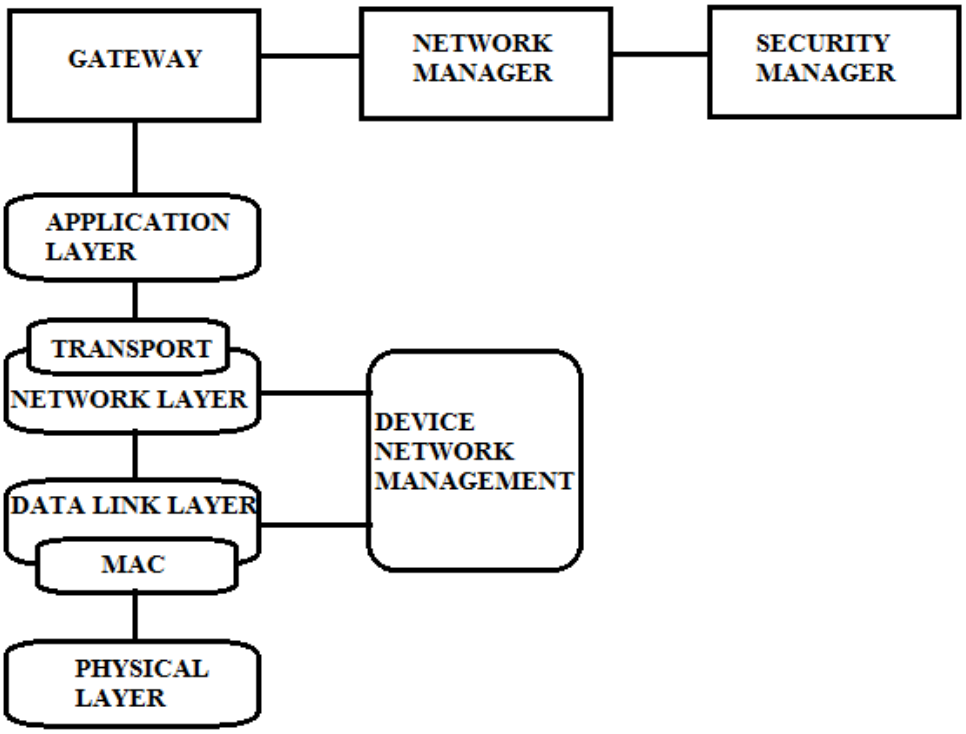

Figura 6. Pilha de Comunicação WirelessHART.

O protocolo WirelessHART é divido em camadas de acordo com o modelo OSI de 7 camadas. As camadas deste padrão são descritas a seguir:

5.1 CAMADA FÍSICA WIRELESHART

A camada física WirelessHART é baseada principalmente no padrão IEEE 802.15.4-2006 2.4GHz DSSS (*Direct Sequence Spread Spectrum*). Esta camada define as características do rádio, como o método de sinalização, potência do sinal, e a sensibilidade do dispositivo. Assim como o protocolo IEEE 802.15.4, o protocolo WirelessHART opera na faixa ISM 2400-2483.5MHz livre de licença, com uma taxa de dados de até 250 kbit/s. Seus canais são numerados de 11 a 26, com um intervalo de 5MHz entre dois canais adjacentes. Estão disponíveis 16 canais na banda de 2450 MHz sendo o centro de freqüência dos canais definido como a seguir [4]:

$$F_C = 2405 + 5(k - 11) \text{ em MHz para k = 11, 12, ..., 26.}$$

Onde k é o número do canal.

Índice	Canal 802.15.4	Freqüência (MHz)
0	11	2405
1	12	2410
2	13	2415
3	14	2420
4	15	2425
5	16	2430
6	17	2435
7	18	2440

Índice	Canal 802.15.4	Freqüência (MHz)
8	19	2445
9	20	2450
10	21	2455
11	22	2460
12	23	2465
13	24	2470
14	25	2475
15	26	não usado

Figura 7. Canais de Freqüência.

O padrão WirelessHART é construído sob o padrão IEEE 802.15.4 com apenas algumas modificações e restrições [3]. Para qualquer dispositivo WirelessHART:

• Há apenas uma ou duas mensagens IEEE 802.15.4 por time *slot* de 10ms (mensagens de difusão, *broadcast*, não possuem *ACK*);

• O tempo mais próximo entre duas mensagens é dentro de um *time slot*, 1ms a partir do final da mensagem, para o início da mensagem de confirmação.

• Todas as mensagens WirelessHART são mensagens do tipo de dados IEEE 802.15.4;

• Somente a banda de freqüência de 2,4 GHz está definida para WirelessHART;

• Os canais 11 a 25 podem ser usados com o padrão WirelessHART. O canal 26, o qual não é permitido em muitas localidades, não é suportado;

Em resumo, a camada física WirelessHART se limita a transmitir e receber mensagens de dados do tipo IEEE 802.15.4. Os itens de destaque na camada física do WirelessHART são:

• Salto de canais:

No WirelessHART o canal físico é alterado a cada transmissão.

• Transmissão de energia:

O padrão IEEE 802.15.4 é definido para uma rede de área pessoal, com espaço operacional de 10 metros. A malha de rede WirelessHART abrange uma área relativamente grande. Todos os dispositivos devem fornecer uma EIRP (*Equivalent isotropically radiated power*) nominal de 10 dBm (10 mW) ± 3dB. A potência de transmissão é programável de -10dBm a +10dBm. A linha máxima de alcance para transmissão deve ser de 100 metros. O componente de hardware para o rádio de um dispositivo WirelessHART são os chips desenvolvidos para o padrão IEEE 802.15.4.

5.1.1 MODULAÇÃO QPSK

A seqüência final codificada é modulada por deslocamento de fase em quadratura (Q-PSK, *Quadrature Phase Shift Keying*). Trata-se de um algoritmo de modulação de fase, onde a fase da onda portadora codifica os bits da informação digital em cada mudança de fase. A modulação O-QPSK é criada através da definição de quatro sinais, defasados de 90 graus. Cada uma das quatro fases possíveis representa dois bits de informação ($2^2 = 4$), ou seja, há dois bits por símbolo. A representação geral de um conjunto de sinais com modulação QPSK é da forma [6].

$$S_{QPSK}(t) = \sqrt{\frac{2E_S}{T_s}} \cos[2\pi f_c\, t + (i-1)\frac{\pi}{2}] \qquad 0 \ll t \ll T_s \qquad i = 1, 2, 3, 4.$$

Sendo T_s é igual a duração do símbolo e tem duas vezes o período do bit e E_s é a energia do símbolo.

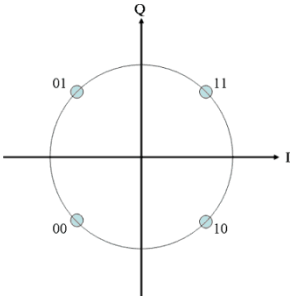

Figura 8 Diagrama de Constelação QPSK

5.1.2 MODULAÇÃO O-QSPK

É a técnica QPSK com *offset*. Essa técnica foi desenvolvida pela necessidade de que os sinais QPSK sejam amplificados apenas por amplificadores lineares, caso contrário há a geração de lobos laterais, levando ao alargamento do espectro ocupado. Porém, amplificadores lineares são menos eficientes. Daí o surgimento da técnica OQPSK, que é menos susceptível aos efeitos de alargamento espectral, permitindo amplificação mais eficiente, através de amplificadores não-lineares [6].

O *offset* vem do fato de que, diferentemente do QPSK, onde os bits dos feixes que serão modulados em fase e em quadratura possuem transição no mesmo instante de tempo; no OQPSK, os bits relativos ao feixe em quadratura sofrem um deslocamento de meio período (metade da duração de um símbolo) no tempo em relação aos bits do feixe em fase [6].

No esquema OQPSK o sinal ocupa a mesma banda ocupada no esquema QPSK, apresentando e utilizando, inclusive, o mesmo espectro. A vantagem é que o esquema OQPSK mantém sua natureza de limitação em banda mesmo após amplificação não-linear, sendo muito atraente para comunicações móveis, onde limitações de banda e uso de amplificadores não-lineares eficientes, para o baixo consumo de energia, são críticos [6] .

5.2 CAMADA DE ENLACE WIRELESHART

Uma característica distinta do padrão WirelessHART é o tempo de sincronização da camada de enlace. Um intervalo de tempo rigoroso de 10ms é definido. É utilizada a tecnologia TDMA para fornecer comunicações sem colisão e determinísticas. O conceito de *superframe* é apresentado como uma seqüência de intervalos de tempo (time *slots*) consecutivos. Um *superframe* é periódico, com o comprimento total do conjunto de *slots* como o período. Todos os *superframes* em uma rede WirelessHART começam com o ASN (*Absolution Slot Number*) 0, no momento em que a rede é criada pela primeira vez. Cada

superframe então se repete ao longo do tempo com base no seu período. No protocolo WirelessHART, uma transação em um intervalo de tempo é descrita por um vetor: {*frame id, index, type, src addr, dst addr, channel offset*} onde o frame ID identifica um *superframe* específico, *index* é o índice do *slot* no *superframe*; *type* indica o tipo do *slot* (transmissão / recepção / inativo); *src addr* e *src dst* são os endereços do dispositivo de origem e do dispositivo de destino, respectivamente; *channel offset* fornece o canal lógico para ser utilizado na transação. Para ajustar o uso do canal, o padrão WirelessHART introduz a idéia de "lista negra" do canal. Canais afetados por consistentes interferências são colocados na lista negra. Desta forma, o administrador de rede pode desabilitar totalmente o uso desses canais. Para suportar o salto de canais, cada dispositivo mantém uma tabela de canais ativos. Devido ao canais na lista negra, a tabela pode ter menos de 16 entradas. Para um determinado *slot* e canal de *offset*, o canal atual é determinado a partir da seguinte fórmula [3]:

Canal atual = (*Offset* do canal + ASN) % Número de canais

O número do canal atual é usado como um índice para a tabela de canais ativos para obter o número do canal físico. Sendo que o ASN está aumentando constantemente, o mesmo deslocamento de canal pode ser mapeado para diferentes canais físicos em *slots* diferentes. Assim, obtém-se a diversidade de canais e aumenta-se a confiabilidade da comunicação. A Figura 9 descreve a arquitetura geral da camada de enlace de dados que é composto por seis módulos principais.

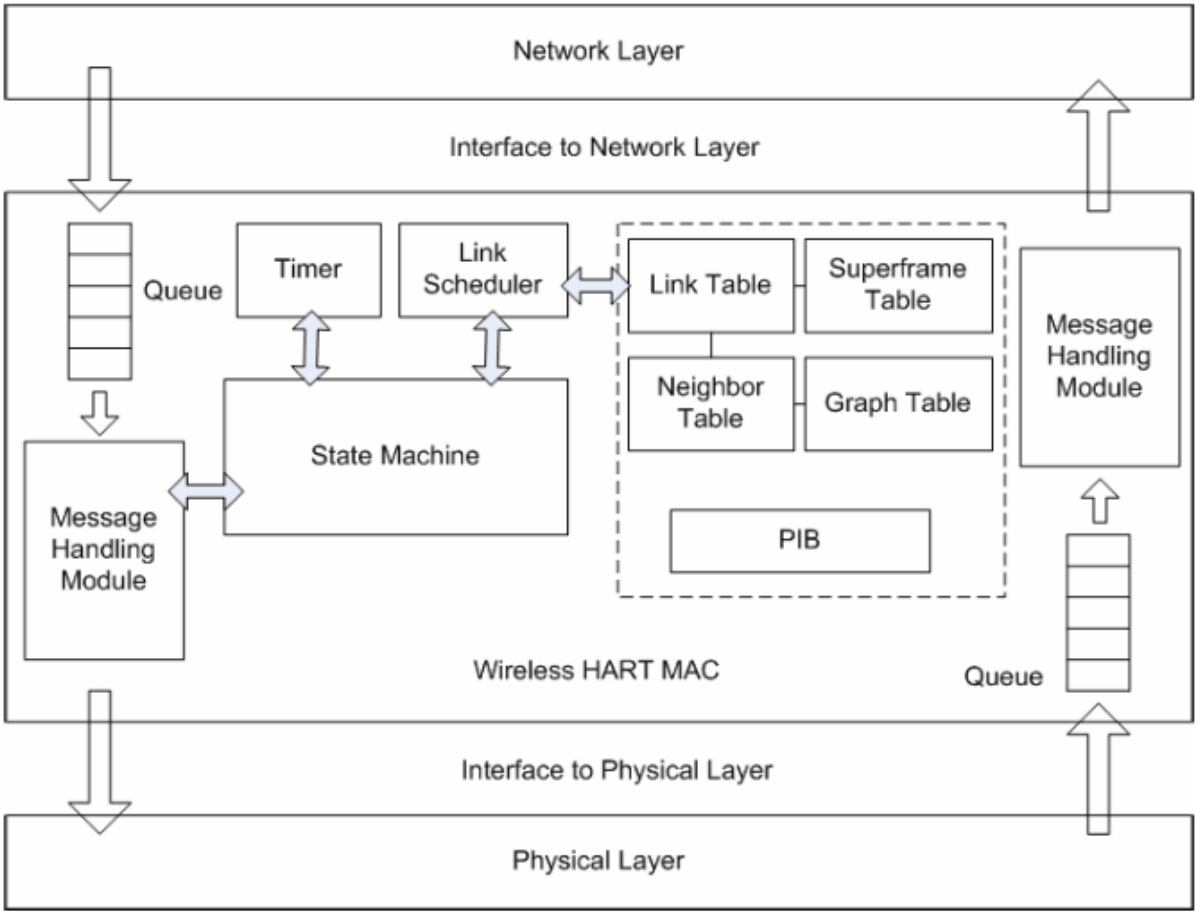

Figura 9. Arquitetura Camada de Enlace WirelessHART.

A camada de enlace tem como tarefa criar e gerenciar os frames utilizados na comunicação, além de ser responsável pela transferência segura de dados entre os nodos da rede, detectando e corrigindo possíveis erros provenientes da camada física. Possui duas sub camadas, LLC e MAC *(Logical Link Control* e *Media Access Control).*

5.2.1 LOGICAL LINK CONTROL

É a mais alta das duas subcamadas da camada de enlace definida no modelo OSI. O LLC é o responsável pelo controle de erros, controle de fluxo de pacotes, montagem dos frames e endereçamento [7].

A estrutura do pacote da camada de enlace DLPDU (*Data-Link Packet* ou *Data-Link Protocol Data Unit*) é composta pelos seguintes campos:

- Um byte ajustado em 0x41;
- Um byte de endereçamento;
- Dois bytes de Network ID;
- Endereços de destino e fonte (podem ser de 2 ou 8 bytes);
- Um byte especifico de DLPDU;
- Campo de dados (*payload*);
- Quatro bytes de integridade da mensagem MIC (*Message Integrity Code*);
- Dois bytes de CRC16 (*Cyclic Redundancy Check*).

A Figura 10 ilustra a estrutura básica do DLPDU [7].

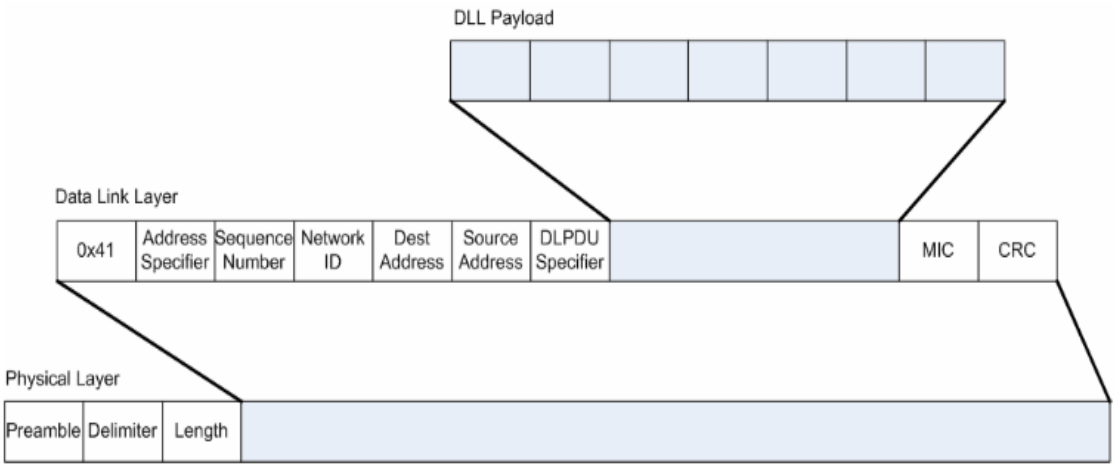

Figura 10 Estrutura Básica da DLPDU

Os campos estão explicados abaixo:

Sequence Number: deve ser ajustado para o byte menos significativo do número de *slot* absoluto (ASN);

Network ID: Sempre são representadas com 2 bytes, sendo o byte menos significativo transmitido primeiro. Se o Network ID não combina com a rede do dispositivo, então não são membros da mesma rede e o pacote é descartado;

Destination e *source address*: WirelessHART pode ter dois tipos de endereços, o apelido com 2 bytes e o IEEE EUI-64 com 8 bytes;

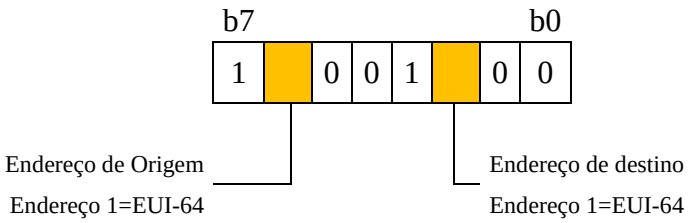

Figura 11 Especificador de Endereço

O apelido (2 bytes) é designado e controlado pelo gerenciador de rede. Por conseqüência, ele é único apenas dentro da rede que pertence. Dois bytes de endereçamento especificam um dispositivo especifico da rede ou deve especificar um endereço de *broadcast* (0xFFFF).

O endereço EUI-64 consiste de 3 bytes provenientes da OUI (*Organizationally Unique Identifier*, gerenciado pela IEEE) e 5 bytes de identificação única controlado pelo protocolo HART [7].

DLPDU specifier: transmitido após o *Network ID* e dos endereços. Os dois bits mais significativos atualmente são reservados. Os próximos dois bits b4 e b5 indicam a prioridade da mensagem sendo "Comando" o com mais alta prioridade e "Alarmes" com a prioridade mais baixa. Bit 3 (b3) indica a chave sendo usada para autenticar a rede. Todos os dispositivos dentro de uma mesma rede devem ajustar esse bit e usá-lo confidencialmente [7]. Esse bit deve estar em *reset* somente quando o dispositivo está em processo de associação,ou seja, enquanto o dispositivo ainda não está autenticado na rede.

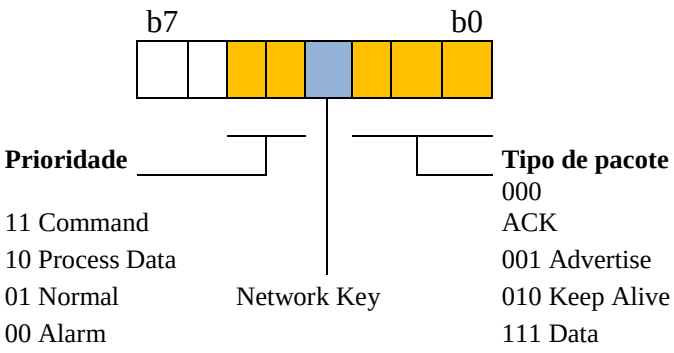

Figura 12 Especificador DLPDU

DLPDU payload: este frame depende do tipo de DLPDU. A DLPDU de dados contém um cabeçalho da camada de rede e o próprio conteúdo do *payload*. DLPDUs de comando da camada de enlace tem conteúdo que dependo do tipo de comando os quais são *Data, Keep Alive, Advertise, Disconnect* e *ACK*.

MIC: (*Message Integrity Code*) é utilizado para autenticação das DLPDUs;

CRC: (*Cyclic Redundancy Check*), baseado no polinômio 16 bit ITU-T CRC. O CRC geralmente é calculado em hardware e utiliza o polinômio a seguir:

$$G_{16} = x^{16} + x^{12} + x^5 + 1$$

5.2.1.1 TIPOS DE DLPDU

Os três bits menos significativos do especificador da DLPDU indicam o tipo de DLPDU que está sendo usada na comunicação.

Existem cinco tipos de DLPDU, os quais estão descritos a seguir:

- *Data:* contém informação da rede e do dispositivo em transito para o destino do seu dispositivo final;
- *Keep Alive*: facilitador da manutenção de conexão entre dispositivos vizinhos. O *payload* para este tipo de DLPDU é vazio e é utilizado para sincronização da rede confirmar comunicação com um dispositivo vizinho. Também na descoberta de um vizinho, o dispositivo deve emitir periodicamente DLPDUs de *keep alive*;
- *Advertise:* fornece informação para os dispositivos vizinhos que podem associar-se com a rede. É utilizado para convidar novos dispositivos a associarem-se na rede. Quando um dispositivo deseja associar-se a rede ele checa por este tipo de DLPDU e usa as informações no *payload* para sincronizar-se com a rede e iniciar o processo de associação;

- *Disconnect:* o DLPDU de desconexão é gerado por um dispositivo que está deixando a rede. Implica que o dispositivo não estará mais disponível para comunicação e pode ser removido da lista de vizinhos. Todos os *links* estabelecidos com os vizinhos deste dispositivo são deletados. Para uma DLPDU do tipo *disconnect* o *payload* está vazio;
- ACK: representa a resposta de nível imediato para receber transmissões DLPDU de outras fontes. ACK DLPDUs são transmitidos por um dispositivo em resposta a uma mensagem *unicast*. O ACK contém o código de resposta que indica se ou não o dispositivo receptor aceitou a DLPDU.

5.2.2 CAMADA DE ACESSO AO MEIO (MAC)

Os principais objetivos da camada MAC são manter o sincronismo dos *slots*, identificar *slots* que necessitam de serviços, escutar pacotes que estão sendo enviados pelos vizinhos e propagar pacotes que vêm da camada de rede [6]. Fundamentalmente a subcamada MAC é responsável pela propagação das DLPDUs através dos *links*. Para realizar esta função, o dispositivo deve possuir:

- Tabelas de vizinhos, *superframes*, *links* e grafos que configuram a comunicação entre o dispositivo e seus vizinhos. Estas tabelas são preenchidas pelo Gerenciador de rede;
- Um agendador de *links* que avalia a tabela de dispositivos e escolhe o próximo *slot* para ser usado na recepção e envio de um pacote;
- Máquina de estados que controla a propagação dos pacotes através da subcamada MAC.

5.2.2.1 TABELAS DE COMUNICAÇÃO

Todos dispositivos mantêm uma série de tabelas que controlam o desenvolvimento da comunicação pelo dispositivo e coletam dados estatísticos das comunicações. Em muitos casos a informação destas tabelas é compartilhada entre as camadas de rede e de enlace. Algumas dessas tabelas são de *superframe*, *links*, vizinhos e grafos. O gerenciador de rede é capaz de configurar múltiplos *superframes* sendo que múltiplos *links* dentro de um *superframe* são configurados para comunicar-se com um dispositivo especifico ou comunicações de transmissão *broadcast*. Cada dispositivo deve suportar múltiplos *superframes*. O *superframe* consiste em um número fixo de *slots*. Ver em Tabela 1 as propriedades do *superframe*.

Tabela 1: Propriedades do *Superframe*.

Conteúdo	Descrição
Superframe ID (Unsigned-8)	Identificação única do *superframe*
Numero de *Slots* (Unsigned-16)	Tamanho do *superframe*
Flags Ativos (Bit-1)	Indica se o *superframe* está ativado
Links[]	Lista de *links* no *superframe*

Os *links* também são alocados pelo *gerenciador de rede*. O *link* inclui uma referência para um vizinho o qual é permitido comunicação com o dispositivo. Esta referência pode ser um dispositivo único ou pode ser para um grupo de vizinhos não especificados (*broadcast*). Cada *link* define uma oportunidade de comunicação sendo que um *link* pode pertencer somente e unicamente a um *superframe*.

Tabela 2: Propriedade dos *Links*.

Conteúdo	Descrição
Link ID	Identificação única do *link*
Identificação de referência do vizinho	Dispositivo vizinho
Tipo do *link* (Enum-3)	{normal, broadcast, join, discovery}

Tx *Link* (Bit-1)	Indica se usado para transmissão
Rx *Link* (Bit-1)	Indica se usado para recepção
Shared *Link* (Bit-1)	Indica se o *link* é usado por mais de um dispositivo
Slot (Unsigned-16)	Número do *slot* no *superframe*
Channel*Offset*(Unsigned-16)	Salto de freqüência de ajuste do canal

As tabelas de vizinhos mantém uma lista de todos vizinhos que podem permitir comunicação com o dispositivo. As tabelas de grafos, são utilizadas na distribuição de mensagens da origem até o destino, sendo que o dispositivo nunca conhece toda a rota de uma mensagem, apenas o próximo salto e o próximo dispositivo o qual deve estabelecer comunicação.

Tabela 3: Propriedade dos Grafos.

Conteúdo	Descrição
Graph ID (Unsigned-16)	Identificação do grafo
Identificação única do destino (Unsigned-40)	Endereço do nó de destino
Nickname do destino (Unsigned-16)	Endereço curto do vizinho
Referência dos vizinhos [ ]	Lista de referências para os vizinhos que estão no próximo salto para o destino

5.2.3 CAMADA DE REDE

A camada de rede e a camada de transporte cooperam para fornecer de forma segura e confiável comunicação de ponta a ponta para os dispositivos da rede. É onde os pacotes recebidos pelos serviços da camada de enlace são transferidos para os dispositivos e os pacotes roteados de outros dispositivos são enviados para a camada de enlace, além de processar os pacotes recebidos da camada de aplicação, ver Figura 13.

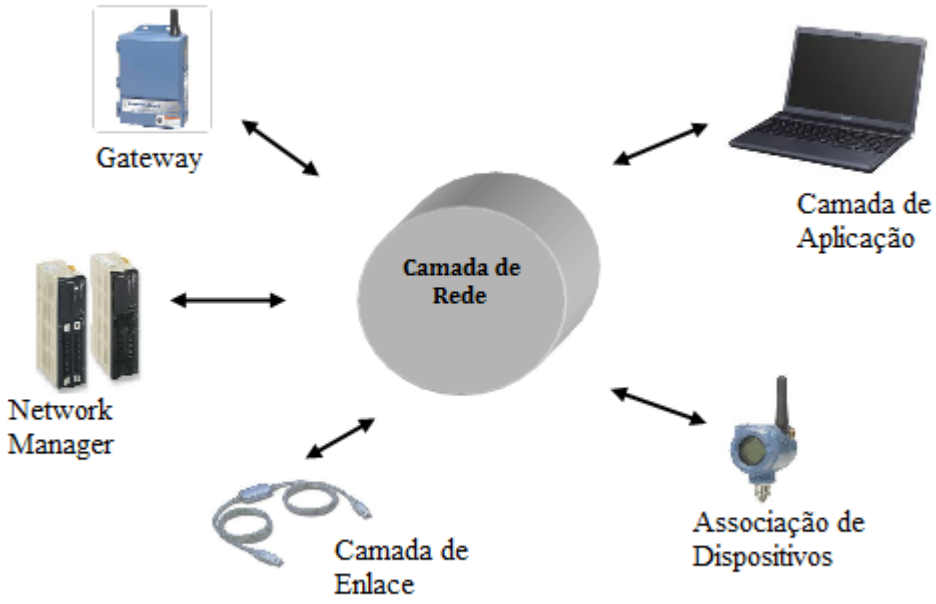

Figura 13. Diagrama de contexto da camada de rede WirelessHART.

No modelo de protocolo de 7 camadas OSI, a camada de rede é a responsável pelas funções de roteamento de rede designando o endereçamento e entrega dos dados. A camada de transporte controla a confiabilidade e tempo de transmissão dos dados entre os nós da rede através de controle de fluxo, segmentação ou separação, e o controle de erro. A camada de sessão controla o dialogo, a sessão, e conexões entre dois nós da rede. No padrão WirelessHART a camada de rede engloba todas estas três camadas [3]. É o ponto de convergência para as tradicionais HART *Token-Passing* e as redes WirelessHART baseadas em TDMA. A Figura 14 descreve a concepção global da camada de rede e de transporte [8].

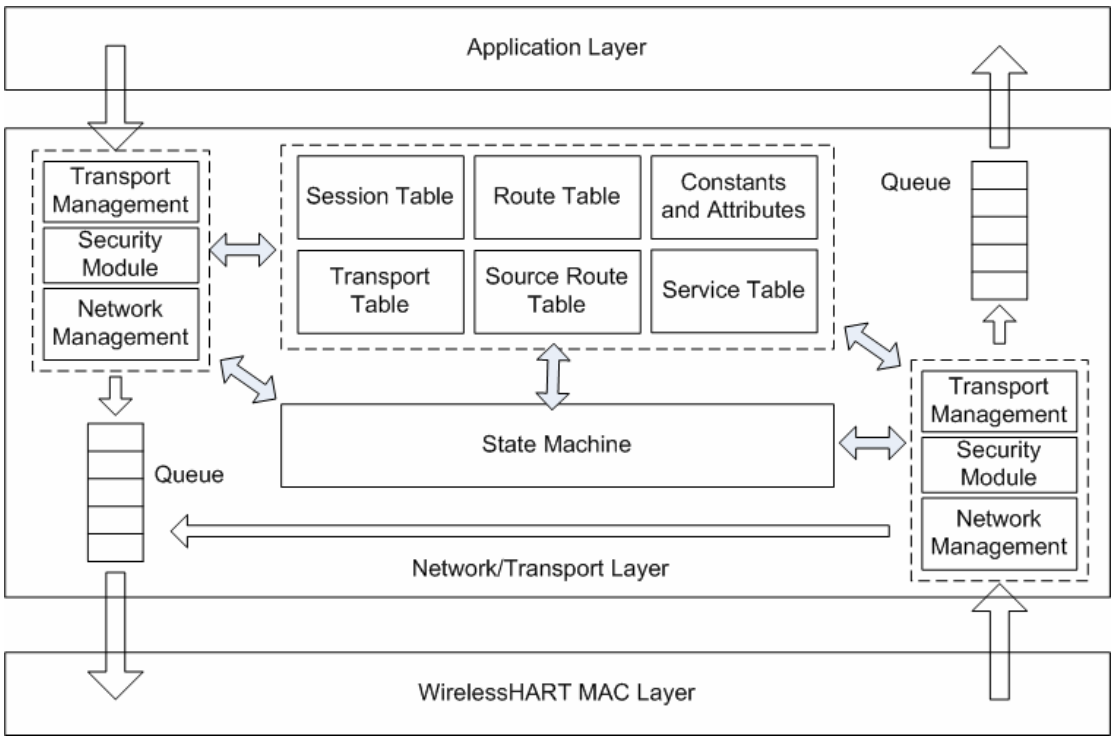

Figura 14. Arquitetura da Camada de Rede e Transporte.

Para dar suporte à tecnologia de comunicação de malha, cada dispositivo WirelessHART deve ser capaz de encaminhar pacotes em nome de outros dispositivos. A camada de rede suporta dois protocolos fundamentais de roteamento definidos no padrão WirelessHART são eles o roteamento por grafos (*Graph Routing*) e o roteamento na fonte (*Source Routing*), dentro destes ainda existem casos especiais como o roteamento por *superframes*. Maiores detalhes sobre o roteamento estão descritos no capítulo 6 deste trabalho.

A fina camada de transporte é especificada como parte da camada de rede que também gerencia as sessões. Acima reside a camada de aplicação HART que define os tipos de dados HART permitidos, procedimentos e comandos. Abaixo deste residem as duas principais estruturas HART: TokenPassing redes com fio e tecnologias de comunicação sem fio TDMA. A Figura 15 mostra o escopo da camada de rede.

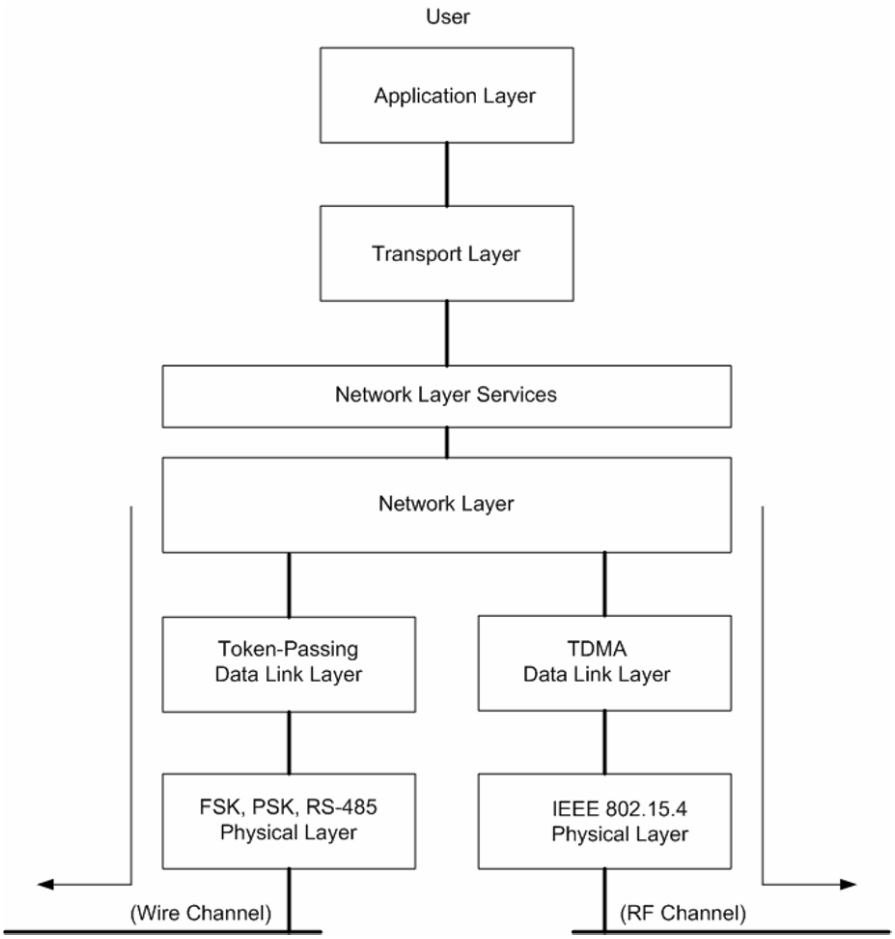

Figura 15. Escopo da Camada de Rede.

O pacote de dados da camada de rede consiste de três funções distintas. Os primeiros campos são referentes à definição da rota da NPDU (*Network Protocol Data Unit*) para seu destino final. Depois os campos de segurança para garantir a comunicação privada sem ser perturbados na comunicação entre os pontos finais da NPDU. Por último os dados da NPDU que são encriptados e contem a informação a ser trocada através da rede (ver Figura 16).

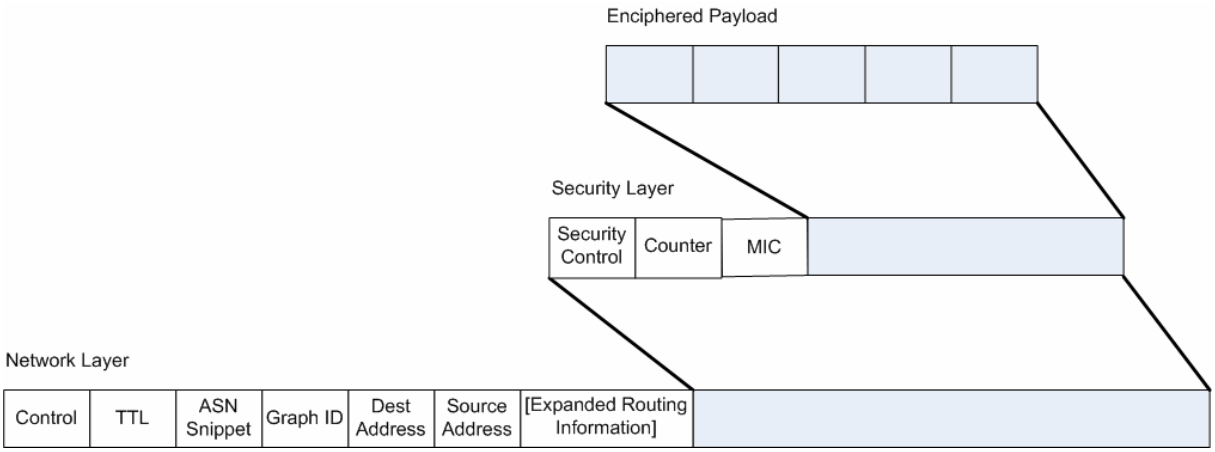

Figura 16 Estrutura da NPDU WirelessHART

Os campos da NPDU contém os seguintes dados:

- Um 1-byte de controle;
- Um 1-byte TTL (*Time To Live*), contador de saltos;
- Os dois bytes menos significativos do Número de *Slot* Absoluto (ASN), contador de latência;
- Dois bytes para o *Graph ID*;
- O endereço do destino final e da fonte;
- Campos de roteamento opcionais.

O *Graph ID* identifica uma lista de nodos qualquer que podem ser usados para enviar um pacote em direção ao destino final.

5.2.3.1 CAMADA DE TRANSPORTE WIRELESSHART

Enquanto a camada de enlace garante que pacotes são transmitidos e propagados com sucesso de um dispositivo para o outro, a camada de transporte é usada para garantir que a comunicação de ponta a ponta seja bem sucedida, ou seja, é responsável por garantir a comunicação através dos múltiplos saltos até o destino final.

5.2.4 OPERAÇÕES DA CAMADA DE REDE WIRELESSHART

Os dispositivos na rede WirelessHART apresentam um avanço através dos diversos estados da rede iniciando no estado *Idle* e continuando até o dispositivo ficar operacional e tornar-se um participante da rede. São seis os principais estados, e são descritos na Tabela 4 [8].

Tabela 4: Definições do Estado da Camada de Rede.

Estado	Descrição
Idle	O dispositivo está quiescente e o transceptor *wireless* não está ativo. Sem conhecimento da rede WirelessHart.
Joining	O dispositivo está escutando pela rede, na tentativa de obter uma publicação e requisitar admissão na rede.
Quarantined	O dispositivo associou-se com sucesso na rede e está conversando com segurança com o network manager. Ainda não é permitido aquisição de dados ou comunicação com o gateway.
Operational	O dispositivo pode ser acessado por aplicações do servidor via gateway.
Suspended	O dispositivo está quiescente. Todas as tabelas da rede estão intactas.
Re-synching	O dispositivo está escutando pela rede. Depois de identificar o tempo do *slot* e o ASN, começará a conectar com os vizinhos novamente.

6 CAMADA DE APLICAÇÃO

O protocolo Hart fornece especificações para facilitar os dois caminhos da comunicação entre o dispositivo de campo e o servidor de aplicações. Os comandos Hart correspondem à camada de aplicação do protocolo de referência OSI [9]. Os comandos dos dispositivos mestre e escravos são a base para a comunicação Hart, sendo que o número do comando determina o conteúdo da mensagem[3]. O número do comando indica a especificação de um único, pacote de dados com uma contagem de bytes fixa.

6.1 PARTIÇÕES DE NÚMEROS DE COMANDOS

Os comandos são alocados em seis grupos que são Universal, Comandos Práticos, Família de Dispositivos, Funções Especificas de Dispositivos e Comandos sem Fio. O protocolo suporta número de comandos de byte único (0-255) e dois bytes estendidos (256-65,535). Comandos 0-255 utilizam o campo de Comandos da mensagem para indicar o número de comando. Para comandos 256-65,535 o campo de comandos apresenta 31 (0x1F) e dois bytes estendidos são encontrados no campo de dados. O conjunto de comandos é divido dentro das seguintes classes:

Comando Universal:

Um conjunto de comandos que devem ser suportados por todos dispositivos compatíveis com o padrão Hart [9];

Comandos Práticos

Um conjunto de comandos para aplicação em uma faixa ampla de dispositivos. Fornece funções comuns para diversos dispositivos, mas não todos. Se um dispositivo utiliza Comandos Práticos, o comando deve ser aplicado exatamente como especificado [9];

Comando não público

Um conjunto especial de comandos (122 a 126) para utilização do fabricante somente durante a fabricação do dispositivo de campo. Por exemplo, o número de identificação do dispositivo, o qual nunca será alterado pelos usuários. Freqüentemente é necessário uma senha para acessar estes comandos. Estes comandos não devem ser utilizados nos dispositivos em campo[9];

Comandos de Família de Dispositivos

Comandos que permitem o ajuste e parametrização dos dispositivos de campo sem necessidade de uso de comandos ou drivers específicos dos dispositivos. Variáveis de dispositivos são classificadas dentro de famílias baseadas no tipo de processo que suportam, por exemplo, pressão, temperatura, atuação em válvulas, vazão, etc. Eles garantem a interoperabilidade entre dispositivos de diferentes fabricantes sem uso individual da descrição do dispositivo. A variável do dispositivo (*Device Variable*) é o objetivo dos Comandos da Família de Dispositivos [9].

Comandos Específicos de Dispositivos

Comandos definidos pelo fabricante de acordo com as necessidades dos dispositivos de campo. Estão na faixa de comandos de 128 a 253. Estes comandos são controlados pelo fabricante do dispositivo[9];

Comandos Wireless

Um conjunto de comandos para equipamentos que suportam o WirelessHart. Todos os produtos que suportam WirelessHart devem ter implementados todos os comandos Wireless.

Tabela 5: Partições dos Comandos HART.

Número do comando	Tipo
0-30, 38, 48	Universal
31	Flag de expansão
32-121 exceto 38 e 48	Comandos práticos
122-126	Não públicos
127	Reservado
128-253	Dispositivo específico
254-511	Reservado
512-767	Comandos práticos adicionais
768-1023	WirelessHART
1024-33,791	Família de dispositivos
33,792-64,511	Reservado
64,512-64,765	Dispositivo específico WirelessHART
64,766-64,767	Reservado
64,768-65,021	Adicional de dispositivo específico
65,022-65,535	Reservado

6.2 REQUISITOS DOS NÚMEROS DOS COMANDOS

A camada de aplicação Hart é baseada em comandos. Para garantir uma implementação consistente, todos os comandos Hart devem seguir os requisitos nos sub-capítulos a seguir. Estes requisitos são aplicáveis para comandos da especificação do protocolo Hart e comandos de dispositivos específicos desenvolvidos por fabricantes de dispositivos compatíveis com HART [9].

6.2.1 AUTONOMIA E REQUISITOS ASSÍNCRONOS

Os comandos Hart devem ser desenvolvidos para serem autônomos e permitir a operação do dispositivo na camada de aplicação, ou seja, nem o mestre ou o escravo devem ser requisitados com prioridades de comunicação de modo a entender a mensagem atual. A resposta deve ser fornecida por um dispositivo escravo sendo clara e unicamente especificada pela requisição do mestre correspondente.

6.2.2 OPERAÇÃO DE COMANDOS

Os comandos Hart devem executar um comando por vez e somente as funções a seguir:

Read: Leitura de dados de um dispositivo de campo. Comandos de leitura não contêm informações no campo de requisição de dados, além das requisitadas pelo dispositivo. Comandos de leitura não podem alterar o modo de operação do dispositivo ou alterar qualquer tipo de dado contido nele.

Write: Escrever dados em um dispositivo de campo. Envia novos valores para serem armazenados no dispositivo. Comandos *write* devem conter os mesmos itens de dados na requisição e na resposta no byte do campo de dados.

Command: Instrui o dispositivo de campo a executar alguma ação (que envolve escrever na memória). Como resultado um comando do tipo *Command* pode ou não ter dados nos bytes de campo *Request/Response* do dispositivo.

6.2.3 COMANDOS CADASTRADOS

Os comandos devem conter índices permitindo o acesso a matrizes ou tabelas de dados armazenados em um dispositivo de campo. O índice é representado por um número inteiro positivo. Isto permite o acesso de um comando único para uma matriz de dados. O número e o tipo dos itens de dados devem ser o mesmo e ocorrer na mesma ordem seqüencial dentro do byte do campo de dados para todos os valores assumidos no índice, ou seja, comandos cadastrados devem definir um pacote fixo de informação onde para cada valor cadastrado a estrutura deste pacote deve ser idêntica.

6.2.4 COMANDOS DE MULTI OPERAÇÃO

Estes comandos permitem um número de sub comando ser colocado dentro do campo de dados do *Request* e *Response* para aumentar o número dos comandos de dispositivos específicos. Comandos de Multi Operação devem ser utilizados somente quando um dispositivo de campo esgota o conjunto permitido de comandos de dispositivo especifico. Todas as operações devem executar a mesma operação de *Read, Write* ou *Command*.

Ao contrário dos comandos cadastrados, o número e o tipo de itens de dados dentro dos byte de *Request* e *Response* podem variar de acordo com o número da operação. Como resultado, as especificações de comando devem incluir um código separado de resposta para comandos específicos em cada operação.

6.2.5 ESTADO DO DISPOSITIVO DE CAMPO

A informação do estado do dispositivo é armazenada no segundo byte de dados no frame do Escravo para o Mestre como uma tabela de bits de campo. O segundo byte de dados indica o status de operação atual do dispositivo de campo como um todo e não sendo associado com a complexidade de nenhum comando. Ao contrário dos requisitos do protocolo Hart (revisão 6.0), este byte deve ser significativo mesmo se um erro de comunicação é notificado no primeiro byte. Na tabela 6 é possível verificar o significado da máscara de bits para o byte do estado do dispositivo.

Tabela 6: Estado do Dispositivo.

Máscara de Bits	**Definição**	
0x80	**Device Malfunction**	O dispositivo detecta um erro ou falha que compromete a operação do dispositivo
0x40	**Configuration Changed**	Uma operação foi realizada alterando a configuração do dispositivo
0x20	**Cold Start**	Falha de potência ou um reset ocorreu
0x10	**More Status Available**	Mais informações disponíveis pelo Comando 48 - Ler informações adicionais

0x08	**Loop Current Fixed**	O loop atual está sendo segurado em um valor fixo e não está respondendo as variações de processo
0x04	**Loop Current Saturated**	O valor de loop alcançou o valor final limite e não pode aumentar e nem diminuir mais
0x02	**Non-Primary Variable Out of Limits**	A variável primária está fora dos limites de operação.
0x01	**Primary Variable Out of Limits**	O PV está além da operação limite

6.2.6 IDENTIFICAÇÃO DO DISPOSITIVO DE CAMPO

Todos os dispositivos devem fornecer algumas informações para os dispositivos mestres quando requisitados. Os dados de identificação, por exemplo, permitem que o mestre lide e determine o conjunto de comandos suportados pelo dispositivo de campo. A seguir uma lista com os dados utilizados na identificação dos dispositivos de campo [9]:

Manufacturer ID: Indica o fabricante do dispositivo. Esta identificação é fornecida ao fabricante pela HCF. Somente o fabricante utiliza esta identificação;

Expanded Device Type: Indica o tipo de dispositivo (nome). Este item indica o conjunto de comandos e dados suportados pelo dispositivo;

Device ID: Um número único e particular. Este número deve ser diferente para cada dispositivo produzido dentro de um tipo;

Device Revision: Um número inteiro indicando o nível de revisão de comandos e dados para o tipo de dispositivo

Software Revision: Um número inteiro (unsigned) indicando a revisão do software do firmware do dispositivo de campo. Um incremento nesse valor é necessário para cada lançamento de uma versão do dispositivo;

Hardware Revision: Um número inteiro (unsigned) indicando a revisão do hardware;

Universal Command Revision: Número inteiro que indica a maior revisão suportada pelo dispositivo;

Private Label Distributor: Indica a companhia que vendeu ou distribuiu o equipamento. Este número deve estar listado nos códigos dos fabricantes;

Tag: Uma etiqueta de 8 caracteres marcada pelo usuário final baseada nas informações de local de uso e aplicação do dispositivo;

Long Tag: Uma etiqueta de 32 caracteres utilizada pelo usuário final.

Estes dados podem ser utilizados para diversos propósitos. Abaixo é possível identificar alguns grupos:

Tabela 7: Aplicação dos Dados de Identificação.

Dados	Propósito
Tag; Tag Long; Private Label Distributor; Device Type	Identificação do usuário do dispositivo de campo
Expanded Device Type e Device ID	Camada de Enlace, Endereço do dispositivo de campo. Quando combinado, identifica um único dispositivo.
Expanded Device Type; Device Revision	Comandos do dispositivo de campo e Conjunto de dados de informação
Device revision; Software Revision; Hardware Revision; Universal Command Revision	Revisão de informação do dispositivo de campo

6.2.7 COMANDOS COM FIO E SEM FIO

O padrão Hart original atribui apenas um byte para os números de comandos, começando com o byte de comando e seguido por um byte com o tamanho e finalmente pelos dados. Para suportar novos comandos e manter a compatibilidade com versões anteriores foi definido um comando especial (31 ou 0x1F em hexadecimal), o qual define que o comando enviado tem dois bytes extras inseridos após o comprimento dos dados [3]. Assim para analisar a mensagem do comando com fio, é necessário ler o primeiro byte para determinar o número do comando. Se este número for igual a 31, então o comando exato deve ser descoberto na seção de dados do comando.

O formato do comando em WirelessHart apresenta em seus dois primeiros bytes o número direto do comando, seguido pelo tamanho dos dados do comando e finalmente pelos dados de comando, sendo possível traduzir a versão com fio para a sem fio. Se o número de comando não é 31, então basta definir o byte de comando alto em zero e copiar os dados do comando. Se o número de comando é igual a 31, copia-se o comando de dois bytes da seção de dados e os dados de comando (os argumentos, índices de tabelas, número de entradas, etc.).

O formato dos comandos de resposta também são diferentes entre as duas versões do protocolo Hart. Primeiramente, há um byte de status e um de status estendido do dispositivo antes de todos os comandos de resposta. Estes são definidos na camada de transporte, mas deverão ser encaminhados para a camada de aplicação. Na versão com fio estes dois bytes não estão presentes, no entanto, a verdadeira diferença está dentro do campo de dados do comando. Há um código de byte de resposta no cabeçalho de cada resposta de comando na versão sem fio enquanto que na versão com fio há um byte de status de dispositivo depois do código de byte de resposta da versão com fio. É necessário utilizar o comando 0 para ler o estado estendido do dispositivo na versão com fio. O analisador de comando quando recebe um pacote de comando precisa verificar se este pertence à versão com fio ou sem fio do Hart.

6.2.8 ESPECIFICAÇÃO DE COMANDOS SEM FIO

Estes comandos são estabelecidos como os requisitos mínimos da camada de aplicação requerida para os dispositivos WirelessHart. A camada de aplicação Hart define os comandos, respostas, tipos de dados, relatório de estado e suporte pelo protocolo, além de certas convenções do Hart que também são consideradas parte da camada de aplicação.

Muitos destes comandos são utilizados para controlar outras camadas dentro da pilha de protocolo como a camada física, camada de dados, camada de enlace e camada de rede.

7 ROTEAMENTO WIRELESSHART

Uma das características associadas às vantagens de confiabilidade e segurança do protocolo WirelessHART é a dinâmica da formação da rede de comunicação entre os diversos dispositivos. Existem três formações básicas de topologias de rede:

- Estrela;
- Malha (*mesh*);
- Estrela e malha ou hibrida.

Na topologia estrela, ver Figura 17, cada dispositivo apresenta-se como a ponta da comunicação onde é capaz de comunicar-se diretamente com o gateway, tendo assim apenas um salto de comunicação. O gateway pode levar a informação então para outros sistemas de interesse. Topologias estrela apresentam alto desempenho e um tempo de latência pequeno. No entanto é necessário que todos os dispositivos participantes estejam no raio de alcance do gateway. Esta topologia é adequada para instalações que necessitam de baixo consumo de energia sobre limitações de alcance geográfico [11].

Figura 17. Topologia de rede estrela.

Em uma rede em malha, ver Figura 18, cada dispositivo pode atuar como um roteador, enviando e recebendo dados de outros dispositivos ou do gateway. Configurações automáticas da rede determinam qual é o melhor caminho para o fluxo dos dados. Este esquema é bom

para as redes de área ampla, com alta redundância, com potência suficiente para todos os participantes encaminharem as mensagens.

Figura 18. Topologia em Malha.

Uma rede em estrela e malha combina as duas topologias já apresentadas sendo que inclui o ganho de velocidade de uma rede tipo estrela com a capacidade de auto-reparação da rede em malha, ver Figura 19. Dispositivos podem ser elementos finais da comunicação assim como roteadores, dependendo da localização deles no sistema.

Figura 19. Topologia de rede tipo híbrida.

O protocolo WirelessHART apresenta dois métodos principais de roteamento de pacotes na rede WirelessHART. São eles o Roteamento em Grafos e o Roteamento na Origem. A seguir é possível verificar uma relação de entidades entre os métodos [12].

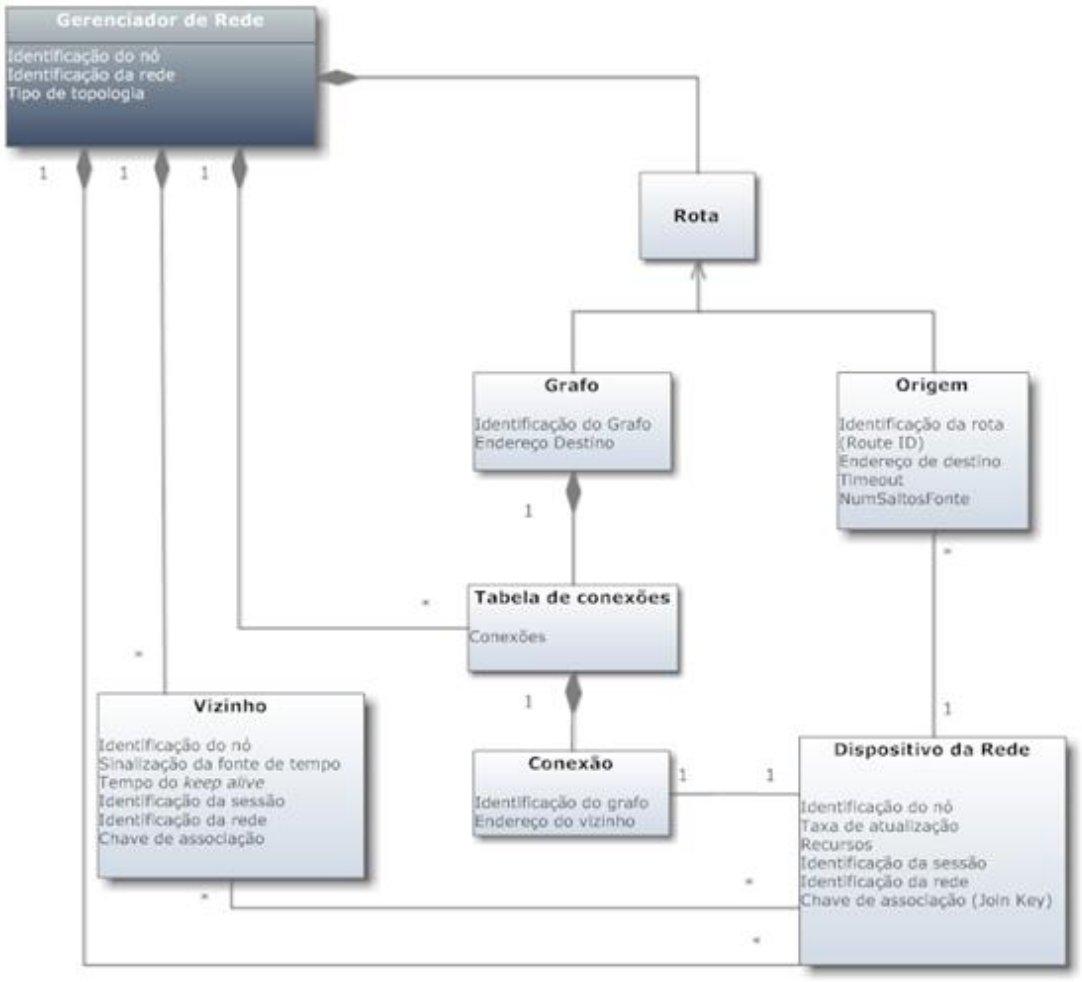

Figura 20. Roteamento da Rede.

Há três componentes básicos na formação da rede WirelessHART: publicação, associação e alocação de recursos. Como parte da publicação, os dispositivos que já fazem parte da rede enviam pacotes anunciando a presença da rede que eles fazem parte. Os pacotes de publicação incluem informações de sincronização de tempo e o identificador único da rede (*Network ID*). Os dispositivos que recebem estes pacotes e possuem o mesmo identificador de rede podem associar-se a rede. Após o dispositivo se associar a rede ele pode negociar com o gerenciador por recursos da rede (informações) [12].

O gerenciador de rede é o responsável por realizar o roteamento das informações dos dispositivos com o objetivo de tornar o roteamento de mensagens eficiente e otimizada. São necessárias informações sobre os requisitos de comunicação e capacidade dos dispositivos na

rede. Após a descoberta destas informações, são ajustadas as conexões até ser obtido um sistema de trabalho adequado. A estratégia de roteamento básica é apresentada abaixo [12]:

7.1 ESTRATÉGIA DE ROTEAMENTO

1) Se há um caminho de apenas um salto para o gateway, deve ser usado;
2) O número mínimo de saltos a ser considerado para a construção de um grafo é dois;
3) O número máximo de saltos a ser considerado quando construído um grafo inicial é quatro;
4) A relação entre a força mais baixa de um sinal em um caminho de dois saltos para a força do sinal em um caminho de um salto correspondente deve ser considerado ao invés de apenas o caminho de um salto;
5) A mesma condição anterior deve também ser considerada para caminhos de 3 e 4 saltos;
6) O limite do nível do sinal para ser usado quando um grafo é construído. Na primeira tentativa, 50% é considerado como ponto inicial. Se nenhum caminho é encontrado usando o nível do limite de sinal especificado, então o limite é diminuído para 75% do valor anterior e o gerador de grafos tenta novamente. Esse recurso mantém até 4 vezes. Se pelo menos uma rota não é encontrada ou ainda não possível, então é considerado que o nó não é alcançável.

O gerenciador de rede contém uma lista completa de rotas, conexões e dispositivos de rede. Quando os dispositivos são adicionados a rede, o gerenciador de rede armazena todas as entradas de vizinhos incluindo informação do nível de sinal reportado de cada dispositivo na rede. Essa informação é utilizada para construir o grafo completo da rede. Uma função importante do gerenciador de rede é configurar os grafos e informações de conexão em cada dispositivo da rede [5].

7.2 ROTEAMENTO EM GRAFOS

Um grafo é uma coleção de caminhos que conectam os nós da rede. Os caminhos em cada grafo são explicitamente criados pelo gerenciador de rede e transferidos para cada dispositivo individualmente. Uma rota de grafos apresenta um conjunto de *links* diretos a dispositivos que podem fornecer rotas de comunicação redundantes entre a origem e o dispositivo de destino final. A rota escolhida como a rota real baseia-se nas condições atuais da rede quando o pacote é transmitido a partir da origem para o destino.

Para enviar um pacote, o dispositivo de origem escreve um *Graph ID* específico (determinado pelo destino) no cabeçalho da rede (NPDU). Todos os dispositivos de rede no caminho para o dispositivo de destino devem ser pré-configurados com as informações do grafo que especifica os vizinhos para os quais os pacotes podem ser enviados.

Todo grafo na rede é associado com um único *Graph ID* sendo necessário incluir esta informação no cabeçalho do pacote para enviá-lo. Um grafo pode especificar mais de um vizinho para ser usado no próximo salto de pacotes seguindo a rota designada. O pacote percorre os caminhos correspondentes ao seu *Graph ID* até alcançar seu destino, ou é descartado. Para ser capaz de rotear pacotes através de um grafo, o dispositivo precisa estar configurado com sua tabela de conexão. A tabela de conexão contém entradas que incluem o *Graph ID* e o endereço de seus vizinhos. Caminhos redundantes são ajustados de modo a obter mais de um vizinho associado com a mesma identificação de grafo. Através do roteamento de grafos, um dispositivo deve executar uma pesquisa pelo *Graph ID* na tabela de conexões e enviar o pacote para qualquer um dos vizinhos listados. Uma vez que qualquer vizinho confirma o recebimento (reconhecimento na camada de enlace) o dispositivo de roteamento pode remover o pacote do seu buffer de transmissão. Caso o reconhecimento de

recepção não seja recebido o dispositivo deve tentar retransmitir o pacote na próxima oportunidade disponível[5].

7.3 ROTEAMENTO NA ORIGEM

O roteamento na origem é um complemento do roteamento em grafos visando o diagnóstico da rede onde é determinado um caminho único dirigido (dispositivos e *links*) entre a origem e o dispositivo de destino. A rota de origem (*source route*) é estaticamente especificada no próprio pacote de dados. No roteamento pela origem é incluído todo o caminho em cada mensagem. Para enviar um pacote ao seu destino, o dispositivo de origem inclui no cabeçalho uma lista ordenada de dispositivos através dos quais o pacote deve percorrer. Como o pacote é encaminhado, cada dispositivo de roteamento utiliza o endereço de rede do próximo dispositivo na lista para determinar o próximo salto até que o dispositivo de destino seja alcançado. Este tipo de roteamento é mais utilizado pelo gerenciador de rede e o gateway que conhecem toda a topologia e constroem os caminhos.

7.4 ROTEAMENTO MISTURADO

O roteamento em grafos e o roteamento na origem são definidos em campos diferentes no cabeçalho da rede e podem coexistir em uma mensagem. Se uma rota chega ao fim, o dispositivo pode tentar outro método [3]. Ainda existem algumas regras especiais que facilitam a passagem das mensagens. Um exemplo, se o endereço de destino é um vizinho, o dispositivo pode ignorar a instrução e enviar a mensagem diretamente para o vizinho (ver Tabela 8) [8].

Tabela 8: Ações de Roteamento.

Roteamento		Endereço de destino		Ação
Origem	Grafos	DLPDU	NPDU	
Sim	Sim	Unicast	Unicast	Envia a NPDU para o próximo endereço do roteamento de origem; ou , se nenhum vizinho combinar com o endereço do Roteamento de Origem, ao longo do roteamento em Grafos.
			Broadcast	Envia NPDU para o próximo endereço do Roteamento de Origem; ou (2)
		Broadcast	não importa	Continua transmitindo para o grafo. Usa o link broadcast no superframe (Graph ID contém a identificação do superframe).
	Não	não importa	Unicast	Envia o NPDU para (1) o próximo endereço do roteamento de origem; ou (2) sinaliza um erro de roteamento da origem.
			Broadcast	Envia o NPDU para (1) o próximo endereço do roteamento de origem.
Não	Sim	Broadcast	não importa	Graph ID é o Superframe ID. Continua transmitindo para o NPDU usando o link broadcast no superframe
		Unicast	Unicast	Envia NPDU ao longo do grafo (utilizando qualquer link normal para o vizinho no grafo), se terminou o Grafo é sinalizado erro no roteamento de grafos
			Broadcast	Envia NPDU para todos os vizinhos no grafo. No fim do Grafo a NPDU é descartada

7.5 ROTEAMENTO POR *SUPERFRAME*

O roteamento por *superframe* (*Superframe routing*) é um caso especial de roteamento em grafos. No roteamento de pacotes por *superframe* os pacotes são atribuídos a um frame e o dispositivo envia a mensagem de acordo com a identificação do *superframe*. Os pacotes são instruídos a seguir a rota do *superframe* da origem para o destino. Todo dispositivo que está associado com qualquer *link* no *superframe* deve receber a informação sobre o *superframe* e o *link*. Supostamente o dispositivo deve selecionar o primeiro *link* disponível do *superframe* e enviar a mensagem, referente ao vizinho o qual ele pertence. Com roteamento por *superframe* a identificação do grafo (*Graph ID*) é definida como a identificação do *superframe* (*Superframe ID*). Se o valor do campo não é superior a 255, então é roteamento pelo *superframe*, se for 256 ou superior, então é roteamento em grafos. Como conseqüência, uma identificação de grafo válida deve ser superior a 255 [3].

7.5.1 COMPARAÇÃO ROTEAMENTO EM GRAFOS E POR *SUPERFRAME*

Roteamento em grafos e por *superframe* apresentam a mesma topologia. No entanto existem algumas diferenças[3]:

- Roteamento em *superframes* tem melhor isolamento do trafego de dados. A entrega dos dados é menos afetada por outros tráfegos na rede, o que garante melhor desempenho para aplicações em tempo real;
- Roteamento em *superframes* necessita de mais recursos do dispositivo. Um dispositivo possui um numero limitado de entradas de *superframe*, ou seja, não é capaz de suportar muitas rotas por *superframes* e ainda precisa de mais computação para localizar o *link* atual com mais *superframes*;
- Ambos são controlados pelo gerenciador de redes. Para ambos os métodos um mau gerenciamento pode causar loops sem fim, caminhos mortos ou destinos inalcançáveis.

Roteamento em *superframe* somente utiliza *links* dentro do *superframe*, roteamento na origem e por grafos podem utilizar *links* em qualquer *superframe*. Como ambos os roteamentos existem, os métodos coexistem e todos os *links* podem ser compartilhados por diferentes tráfegos de dados. Caso somente seja utilizado roteamento em *superframes*, então teremos *links* dedicados, os quais não devem interferir em quaisquer outros tráfegos de dados

8 SOFTWARE PARA ANÁLISE DE ROTEAMENTO

Através do estudo das principais características do protocolo WirelessHART, este trabalho propôs o desenvolvimento de uma ferramenta de software capaz de obter algumas informações com as características do comportamento da rede e do estado dos dispositivos como, por exemplo, dispositivos que estão na rede, entre quais dispositivos há comunicação (topologia), tipo de *links* entre eles, presença de grafos, qualidade do sinal das comunicações e ainda informações adicionais dos dispositivos como (fabricante, modelo, apelido, endereço do dispositivo, função entre outras) informações que de outra maneira os administradores da rede teriam que procurar através de meios mais descentralizados.

8.1 HART SOBRE UDP

O aplicativo desenvolvido utiliza o protocolo UDP para realizar a comunicação com o Gateway WirelessHART[15]. O Gateway por sua vez comunica-se com os dispositivos na rede, ver Figura 21.

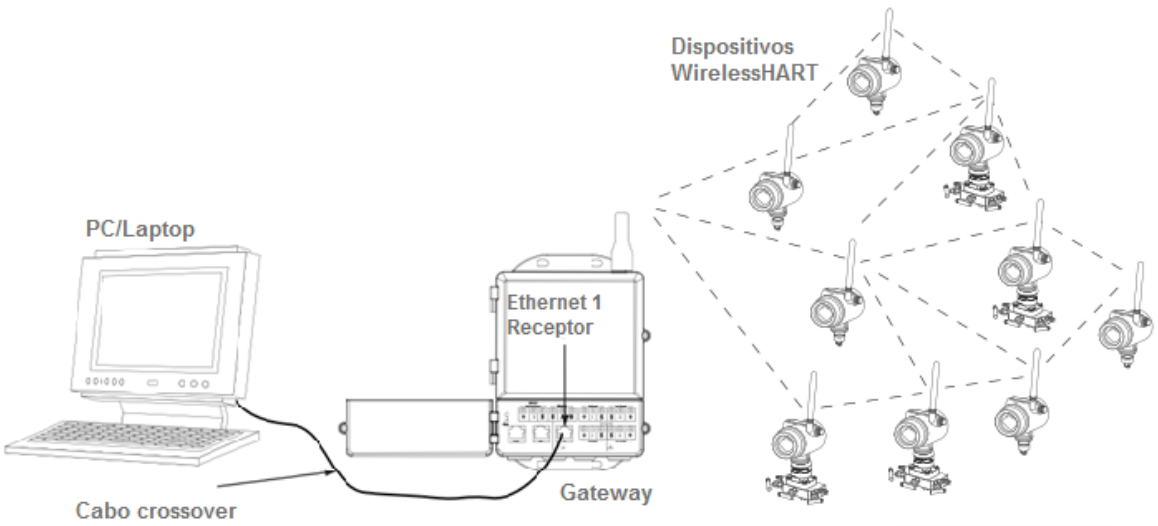

Figura 21. Ilustração da Conexão entre o PC e o Gateway WirelessHART.

Através do protocolo Hart sobre UDP foi possível construir toda a aplicação para monitoração e controle dos dispositivos WirelessHART. Na figura 22 é possível verificar o formato geral do frame Hart sobre UDP.

Versão	tipo	msgID	Status	seq. H	seq. L	tam. H	tam. L	*payload*

Figura 22. Frame UDP.

A descrição dos itens está a seguir:

Versão

Versão do protocolo. Atualmente é a versão 0x01.

Tipo

Identifica a direção, se é requisição ou resposta.

Tabela 9: Tipo de Mensagem.

Tipo	Descrição
0x00	Requisição
0x01	Resposta

MsgID

Identificador da mensagem. Indica se é um procedimento de estabelecimento de conexão ou troca de dados com o servidor. A interpretação ou não do pacote de dados é função do identificador de mensagem.

Tabela 10: Identificador de Mensagem.

msgID	Descrição	Operação
00	Open	Abre conexão UDP
01	Close	Termina conexão UDP
02	Keepalive	Mensagem periódica para identificar conexão ativa.
03	PDU	Mensagens de HART

Status

Indica o estado da comunicação. Nas requisições deve ser colocado em zero (0x00). Nas mensagens de resposta, indica o resultado da comunicação. O valor zero (0x00) indica comunicação com sucesso, sem erros.

Seqüência parte alta e baixa

Número da seqüência da mensagem, expresso em 16 bits, no caso da PDU de dados. As mensagens de resposta têm o mesmo número de seqüenciamento da requisição, permitindo ao cliente da requisição realizar a associação da resposta com a requisição realizada (protocolo UDP não possui seqüenciamento de mensagens).

Tamanho parte alta e baixa

Comprimento total do datagrama. O comprimento da mensagem é contado a partir do byte da versão até o último byte do *payload*.

No Figura 23 há uma representação de troca de mensagens entre o PC e o gateway baseado no HART sobre UDP.

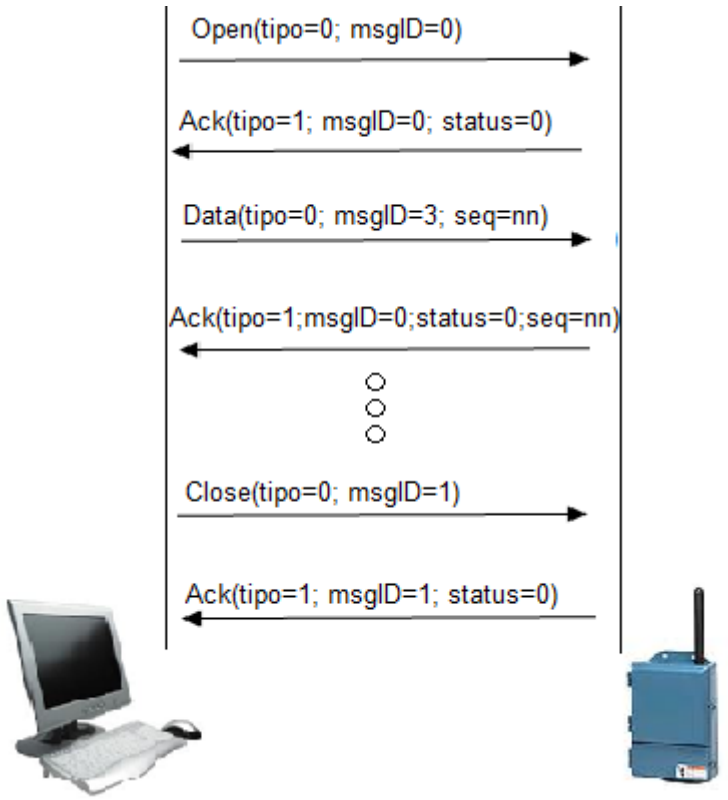

Figura 23. Diagrama de Comunicação HART sobre UDP.

Payload

O campo de dados da mensagem é relativo ao campo do tipo. Para as mensagens de *open/close* e *keep alive*, possuem informações relativas ao controle da conexão, como timeouts e tempos de *keep alive*. Para as mensagens de dados , PDUs *(Protocol Data Unit*), o *payload* contém uma mensagem no formato HART.

O formato das mensagens HART são diferentes para as requisições e respostas. As respostas incluem dois bytes de código de resposta e estado, que não estão presentes nas requisições.

Tabela 11: Dados de requisição HART sobre UDP.

Requisição		
Byte	***Offset***	**Descrição**
00	0x08	Delimitador HART (0x82 – Requisição com endereços de 5 bytes (UID Hart), master to slave)
01-05	0x09	UID – Identificador único HART do destinatário da mensagem.
06	0x0E	Comando HART – Para comandos expressos em dois bytes (extended command number), este campo contem o comando 31(0x1F).
07	0x0F	Tamanho do frame HART, O tamanho é contado a partir do próximo byte excluindo o byte de verificação.
08-09	0x10	Expanded command number. Número de comando expresso em 16 bits (quando o byte 06 for diferente de 0x01F).
10	0x12--	Argumentos do comando HART. (O número de bytes varia de zero a n, em função do comando utilizado).
(08+tam)	(0x10+tam)	Checksum (Ou-exclusivo dos bytes do frame, desde o delimitador até o último byte dos argumentos do comando).

Os dois bits mais significativos do byte 01 não reproduzem o Identificador Único e sim identificam o tipo de mestre (principal ou secundário) e se a mensagem é do modo *burst* (emissão periódica da mensagem).

B1.7 = Mestre (1, primário; 0, secundário);

B1.6 = Burst mode – para mensagens HART sobre UDP este bit sempre será 0.

O endereço padrão de um gateway WirelessHART é 0xF981002, mas com as regras de formação do Identificador Único do dispositivo a ser enviado o endereço presente na mensagem será 0xB981002. Para o endereço de um dispositivo aplicam-se as mesmas regras, no caso de um dispositivo com identificador único de 0xE0FF00FA terá na mensagem o identificador 0xA0FF00FA.

Nas respostas o dispositivo HART escravo retorna com os dados enviados pelo mestre, junto com dois bytes de estado e código de resposta do comando assim como os dados de resposta.

Tabela 12: Dados de Resposta HART sobre UDP.

Resposta		
Byte	***Offset***	**Descrição**
00	0x08	Delimitador HART (0x86 – Requisição com endereços de 5 bytes (UID Hart), slave to master).
01-05	0x09	UID – Identificador único HART do destinatário da mensagem. Para comandos direcionados ao gateway, o endereço será 0xB981000002 (UID padrão para gateways WiHart).
06	0x0E	Comando HART – Para comandos expressos em dois bytes (extended command number), este campo contem o comando 31 (0x1F).
07	0x0F	Tamanho do frame HART, O tamanho é contado a partir do próximo byte excluindo o byte de verificação.
08	0x10	Código de resposta do comando/erro de comunicação. Quanto o msb estiver em um (>=0x80), indica erro de comunicação e os demais bits indicam o erro. Quando msb estiver em zero (< 0x80), indica o código de resposta específico do comando (ver descrição de cada comando HART). Hart
09	0x12--	Estado do dispositivo. Hart Technical
10-11	0x12	Expanded command number. Número de comando expresso em 16 bits
12--	0x13	Argumentos do comando HART. (O número de bytes varia de zero a n, em função do comando utilizado).
(08+tam)	(0x10+tam)	Checksum (Ou-exclusivo dos bytes do frame, desde o delimitador até o último byte dos argumentos do comando).

Na Figuras 24e 25 estão os *frames* registrados através do software Wireshark para requisição e resposta do comando 814.

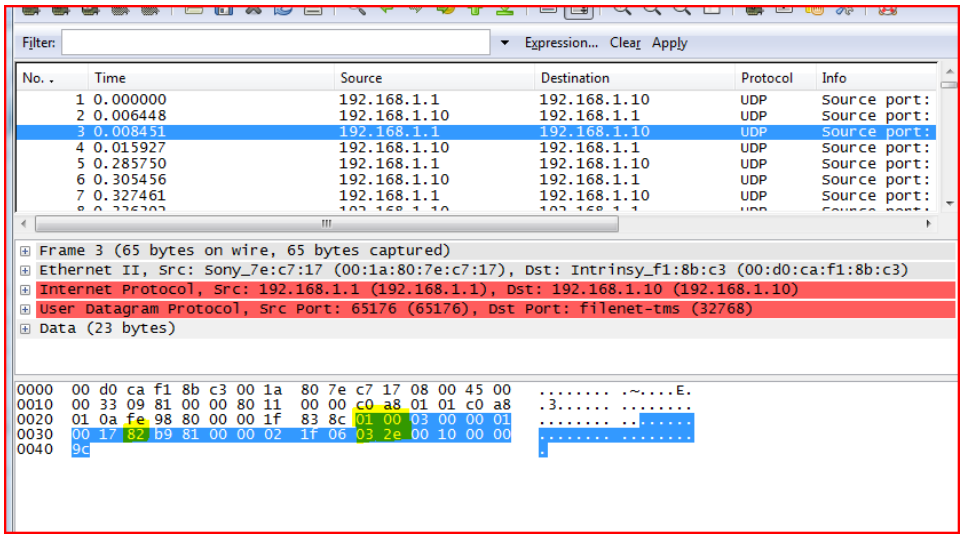

Figura 24. Requisição do comando 814.

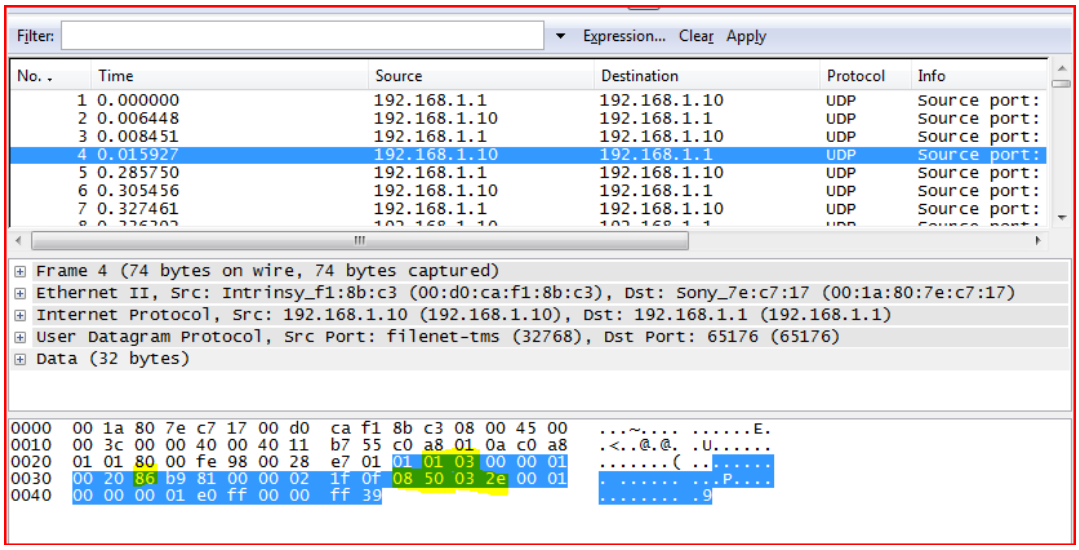

Figura 25. Resposta do comando 814.

Nas Figuras 26 e 27 respectivamente estão em maiores detalhes os campos do comando 814 do protocolo WirelessHART requisição e resposta.

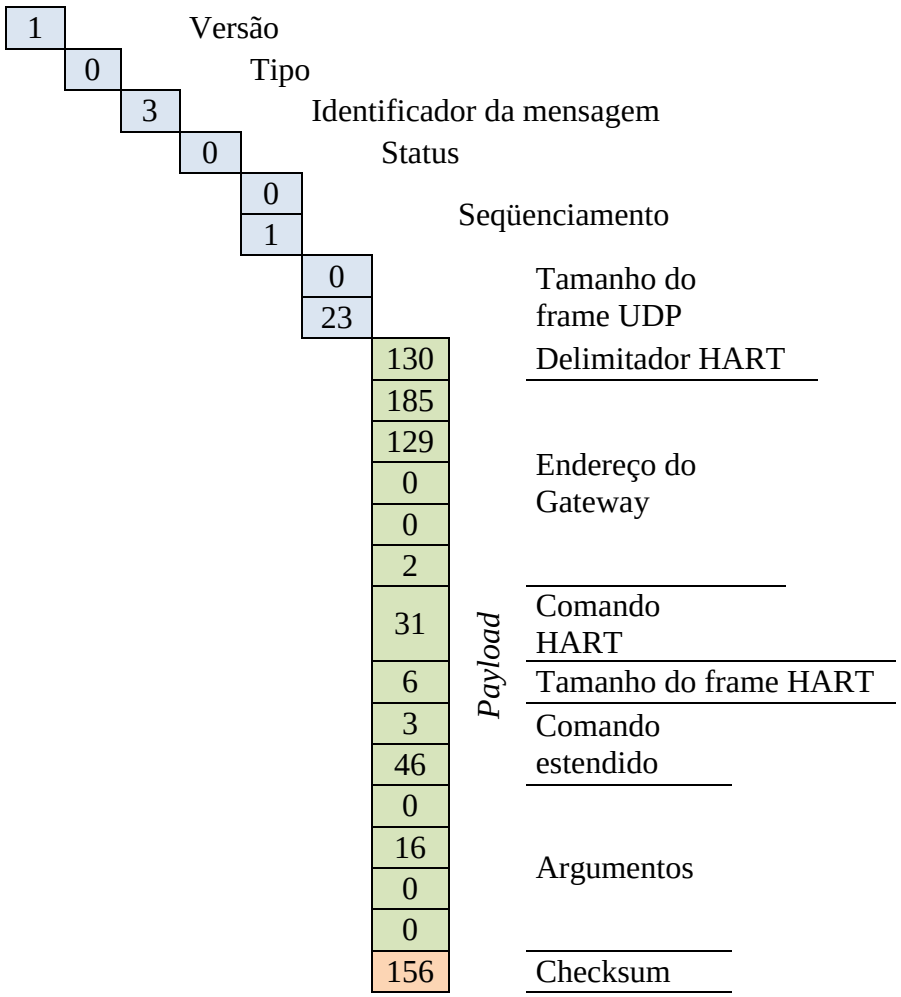

Figura 26. Identificação dos campos do datagrama requisição do comando 814.

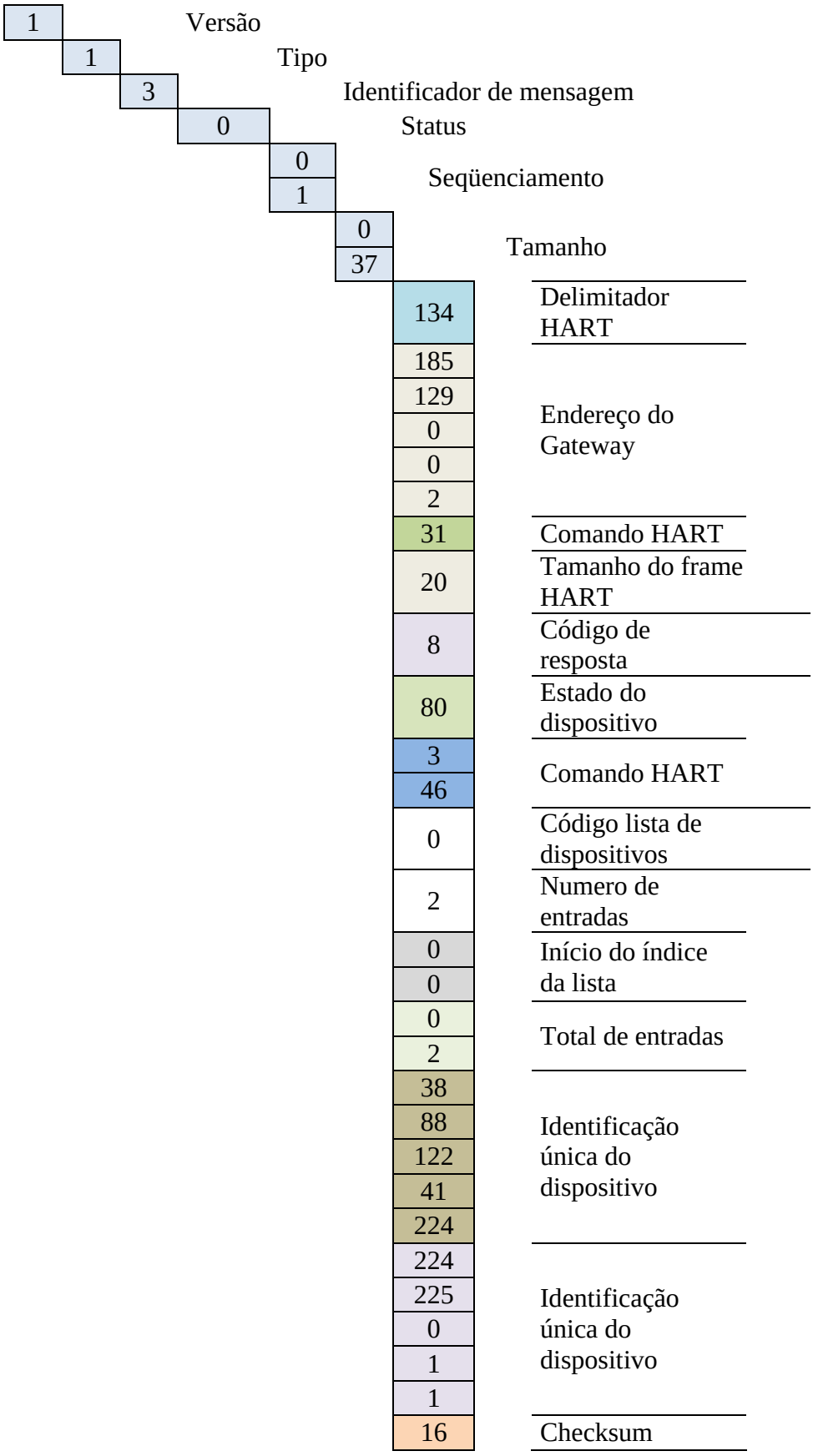

Figura 27. Identificação dos campos do datagrama UDP resposta do comando 814.

8.2 COMANDOS UTILIZADOS

Da mesma forma que o protocolo WirelessHART é baseado em comandos, a aplicação desenvolvida por conseqüência é totalmente dependente do uso dos comandos para a obtenção dos dados desejados para análise da rede e dos dispositivos. Os comandos HART são encapsulados sobre UDP e enviados para o Gateway que responde a requisição enviando os dados solicitados. Ver no Anexo A tabelas com os dados de requisitos e resposta dos comandos implementados[10]. Abaixo um breve detalhamento dos comandos utilizados:

8.2.1 COMANDO 780 – *REPORT NEIGHBOR HEALTH LIST*

O comando 780 é um comando de aplicação sem fio. Este comando fornece dados estatísticos dos vizinhos que apresentam *link* com o dispositivo. As variáveis de maior interesse no comando 780 são:

- Número total de vizinhos: Informa número de dispositivos que possuem *link* com o dispositivo interrogado;
- Apelido dos vizinhos;
- RSL, *Receive Signal Level*, da origem para o destino: Informa o nível de sinal recebido em dBm;
- Pacotes transmitidos para vizinhos;
- Pacotes recebidos do vizinho;
- Sinalização do vizinho:

  0x01 *Time source*;
  0x40 *Keep alive pending*;
  0x80 *Path failure (read only).*

8.2.2 COMANDO 781 – *READ DEVICE NICKNAME*

Um endereço na rede do dispositivo de 16 bits. O gerenciador de rede atribui uma identificação única para cada dispositivo da rede.

8.2.3 COMANDO 782 – *READ SESSION LIST*

Através deste comando é possível obter informações sobre as sessões estabelecidas entre os dispositivos e o gerenciador de rede. Sessões são endereçadas pela suas posições no dispositivo. Por segurança, uma sessão WirelessHART é orientada e todos os dispositivos devem suportar múltiplas sessões. Uma sessão permite uma comunicação privada e segura entre um par de endereços na rede. Após o dispositivo associar-se a rede geralmente são ajustadas quatro sessões no dispositivo.

- Uma sessão é *unicast* entre o gerenciador de rede (F980) e um dispositivo, quando o gerenciador de rede gerencia o dispositivo;
- Entre o gateway (F981) e o dispositivo (*unicast*), são as comunicações normais (por exemplo, dados de processo);
- Gateway *broadcast* (para todos os dispositivos na rede);
- Gerenciador de rede *broadcast* (para todos dispositivos na rede). Todos dispositivos na rede tem a mesma chave de rede. Esta sessão é utilizada globalmente para gerenciar todos dispositivos na rede.

Quando uma sessão nova é criada o parâmetro *Nonce Counter Value* do dispositivo é ajustado para zero. Na camada de rede, cada sessão tem uma chave diferente para encriptar e autenticar pares de dispositivos. Chaves de sessões são geradas pelo gerenciador de rede e são únicas para cada comunicação ponta a ponta entre dois dispositivos [5]. Chaves *Unicast* e *broadcast* são muito diferentes em como elas armazenam e usam o *nonce counter value*. Um dispositivo deve ter mais de uma sessão definida por par de dispositivos as sessões podem ser estabelecidas entre quaisquer dois dispositivos na rede.

8.2.4 COMANDO 783 – *READ SUPERFRAME LIST*

Este é um comando da camada de enlace e permite buscar informações sobre um *superframe* atribuído a um dispositivo. Um dispositivo pode participar de um ou mais *superframes* simultaneamente, mas não todos dispositivos precisam participar em todos os *superframes*. Se um dispositivo for configurado para participar em múltiplos *superframes* de diferentes tamanhos é possível estabelecer diferentes cronogramas de comunicação e matrizes de conectividade que podem trabalhar ao mesmo tempo.

- Número de *superframes* ativos;
- *Superframe* ID;
- Número de *slots* do *superframe*: Determina com que freqüência cada *slot* é repetido, estabelecendo assim um cronograma de comunicação para os dispositivos que utilizam os *slots*;
- Sinalização do *superframe*:

0x01 Ativo;

0x80 Handheld *superframe*.

8.2.5 COMANDO 784 – *READ LINK LIST*

A rede WirelessHART contém um cronograma geral o qual é criado e administrado pelo gerenciador de rede[7]. O agendamento é organizado em *superframes*. Cada *superframe* tem uma divisão interna de *slots* chamados de *slots* relativos.

Os *links* estão associados com um dispositivo especifico, para cada *link* há *slots* alocados para um ou mais dispositivos. Os *links* estão endereçados pela sua posição na lista de *link* dos dispositivos e não possuem uma implementação particular. Os principais dados obtidos deste comando estão listados abaixo:

- *Superframe* ID: Identifica um *superframe* especifico
- Número do *slot* do *superframe* para determinado *link*;
- Canal utilizado: Indica o canal lógico utilizado na transação;

- Vizinho: Apelido do vizinho associado para este *link* ou 0xFFFF se for *broadcast*;
- Opções do *link*: Cada *link* possui exclusivamente um *slot* associado a um *superframe* com as opções *(transmit, receive, shared*) e os tipos;
- Tipo do *link*: (normal, advertising, discovery e join).

8.2.6 COMANDO 787 – *REPORT NEIGHBOR HEATH LIST*

Comando 780 e 787 no modulo de rádio enviam o mesmo resultado. Nos dispositivos da Emerson (sensor temperatura 648), o comando 787 possui restrição de acesso. Na especificação HART o comando 787 identifica os vizinhos descobertos que não estão ligados a este nodo.

8.2.7 COMANDO 800 – *READ SERVICE LIST*

Uma camada de rede inferior fornece um conjunto de aplicações para a camada superior, essas aplicações são chamada de serviços. Os seguintes dados podem ser extraídos deste comando:

- Número de serviços ativos
- Identificação de Serviço: *Service ID*;
- Sinalização de requisição de serviço:

  0x01 Source;

  0x02 Sink;

  0x04 Intermediate.

- Domínio de aplicação de serviço:

  0 Publish;

  1 Event;

  2 Maintanance;

  3 Block transfer.

- Apelido do par com o qual o serviço é requisitado:
- Periodo:
- Route ID.

8.2.8 COMANDO 802 – *READ ROUTE LIST*

Este comando faz parte do grupo dos comandos da camada de rede e identifica algumas variáveis de interesse para determinar os caminhos dos pacotes na rede.

- Número de rotas ativas;
- Origem e destino;
- Route ID: Identificador para uma rota especifica;
- Graph ID: Identificador de um grafo especifico;
- Flag de Source Route: (0, não utiliza source route; 1, utiliza).

8.2.9 COMANDO 840 – *READ NETWORK DEVICE´S STATISTICS*

Este comando retorna o número de grafos, frames e *links* ativos que o dispositivo apresenta. Esta informação o gateway tem disponível e pode responder imediatamente.

- Apelido do dispositivo solicitado;
- Grafos Ativos;
- Frames Ativos;
- *Links* Ativos;
- Número de vizinhos;
- Latência de comunicação: o tempo que um pacote leva para atravessar uma conexão na rede;
- Número de joins;
- Data último join;
- Pacotes terminados no dispositivo;
- Pacotes gerados no dispositivo;
- Falhas enlace;
- Falhas na rede;

- Erros CRC;
- Nonce counter number by;
- Nonce counter number from;
- Desvio padrão de Latência.

8.3 ESTRUTURA DO SOFTWARE

O aplicativo foi desenvolvido utilizando o ambiente de desenvolvimento da Microsoft Visual Studio 2008[14], ver Figura 28. O Visual Studio suporta linguagens de programação como: C, C++, C#. Para esta aplicação foi utilizado a linguagem C.

A aplicação foi desenvolvida utilizando dois projetos hartcli.cpp e hartip.cpp dentro do ambiente do Visual Studio. O projeto hartcli comporta a interface do programa e a formatação dos dados recebidos enquanto a hartip gera uma DLL (dynamic *link* library) e comporta as funções de montagem dos frames sobre UDP, as funções de envio de mensagem, validação de dados recebidos, estruturas dos comandos. Na Figura 29 é possível verificar tela de console do início da aplicação.

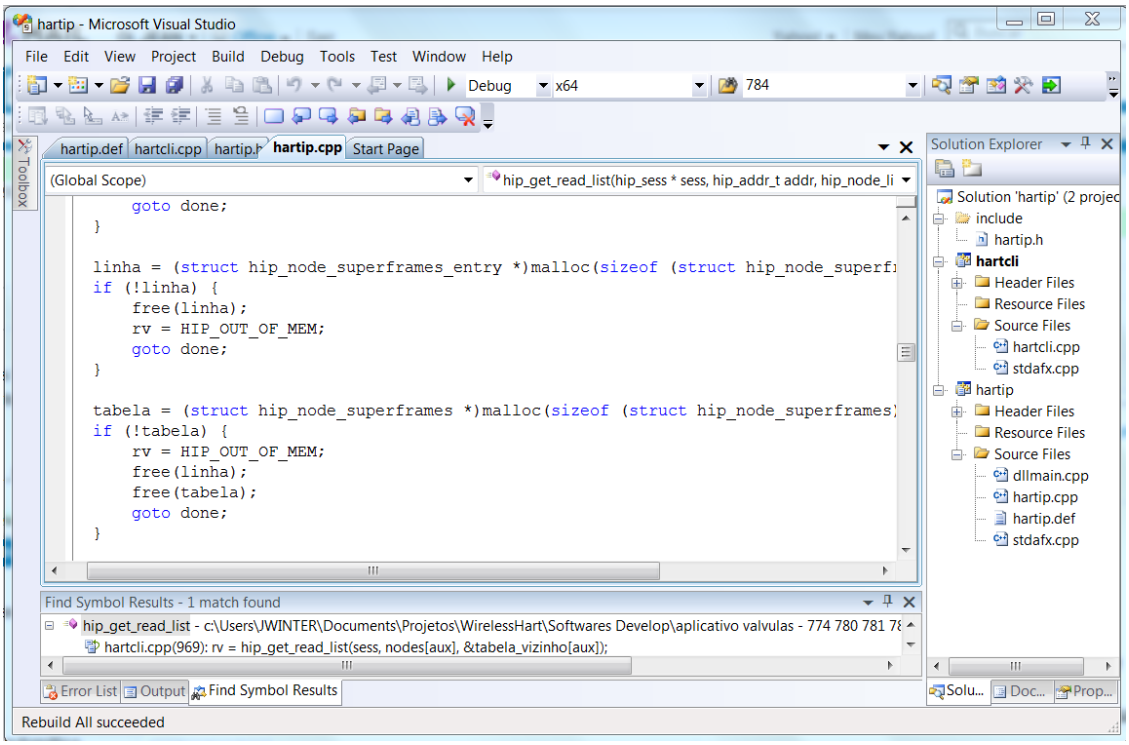

Figura 28. Ambiente de Desenvolvimento.

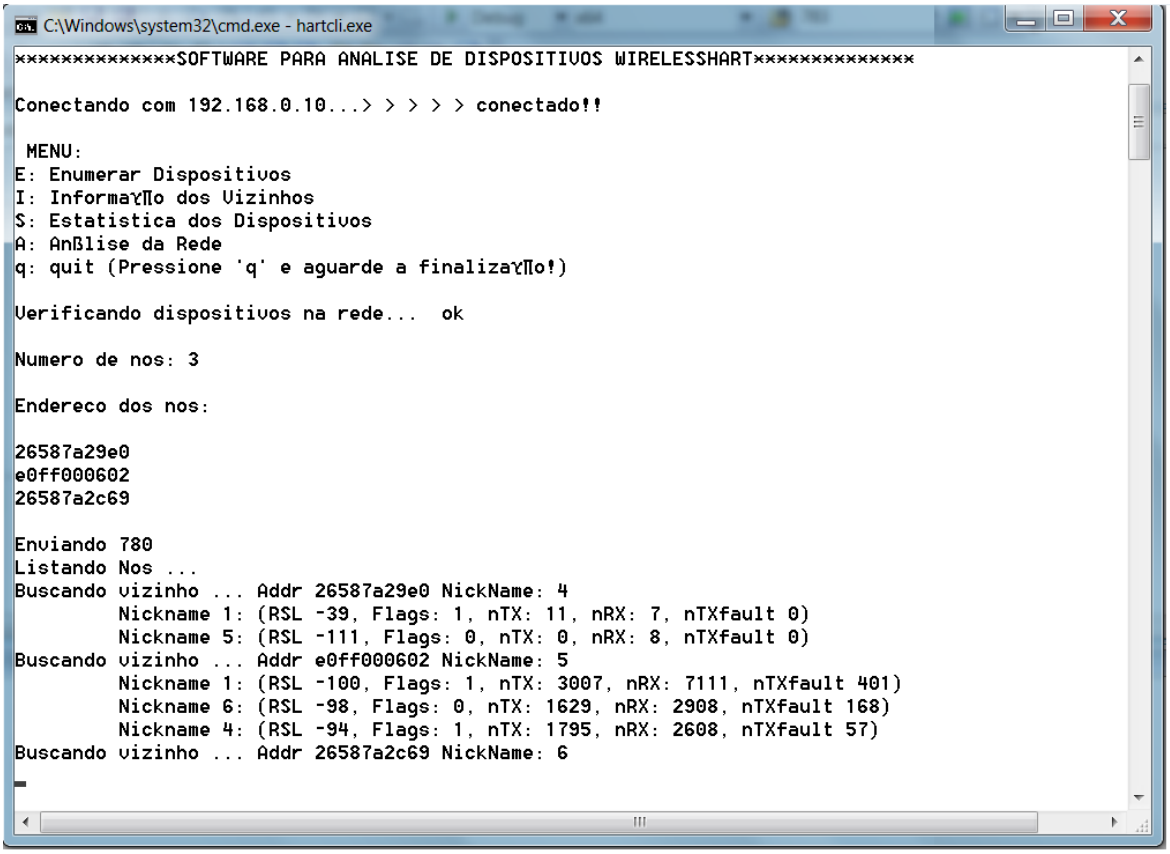

Figura 29. Menu do Aplicativo.

Nos fluxogramas a seguir estão descrito as principais funções utilizadas na aplicação associadas aos comandos do protocolo WirelessHART. Alguns comandos foram implementados no programa de modo a complementarem a obtenção dos dados que são utilizados como argumentos de outros comandos.

A Figura 31 representa as funções da primeira tela do software, onde as opções são apresentadas ao usuário. Para executar o aplicativo é necessário entrar com o endereço IP do gateway desejado para assim estabelecer a comunicação. No caso, como havia disponibilidade de apenas um gateway o endereço foi pré-configurado e mantido fixo em 192.168.0.10. As seguintes opções são apresentadas:

i. Enumera dispositivos;
ii. Informação dos vizinhos;
iii. Estatística dos dispositivos;
iv. Análise da rede.

A Figura 30 representa os símbolos utilizados nas Figuras 31, 32, 33, 34, 35, 36 e 37.

Figura 30. Legenda dos Fluxogramas.

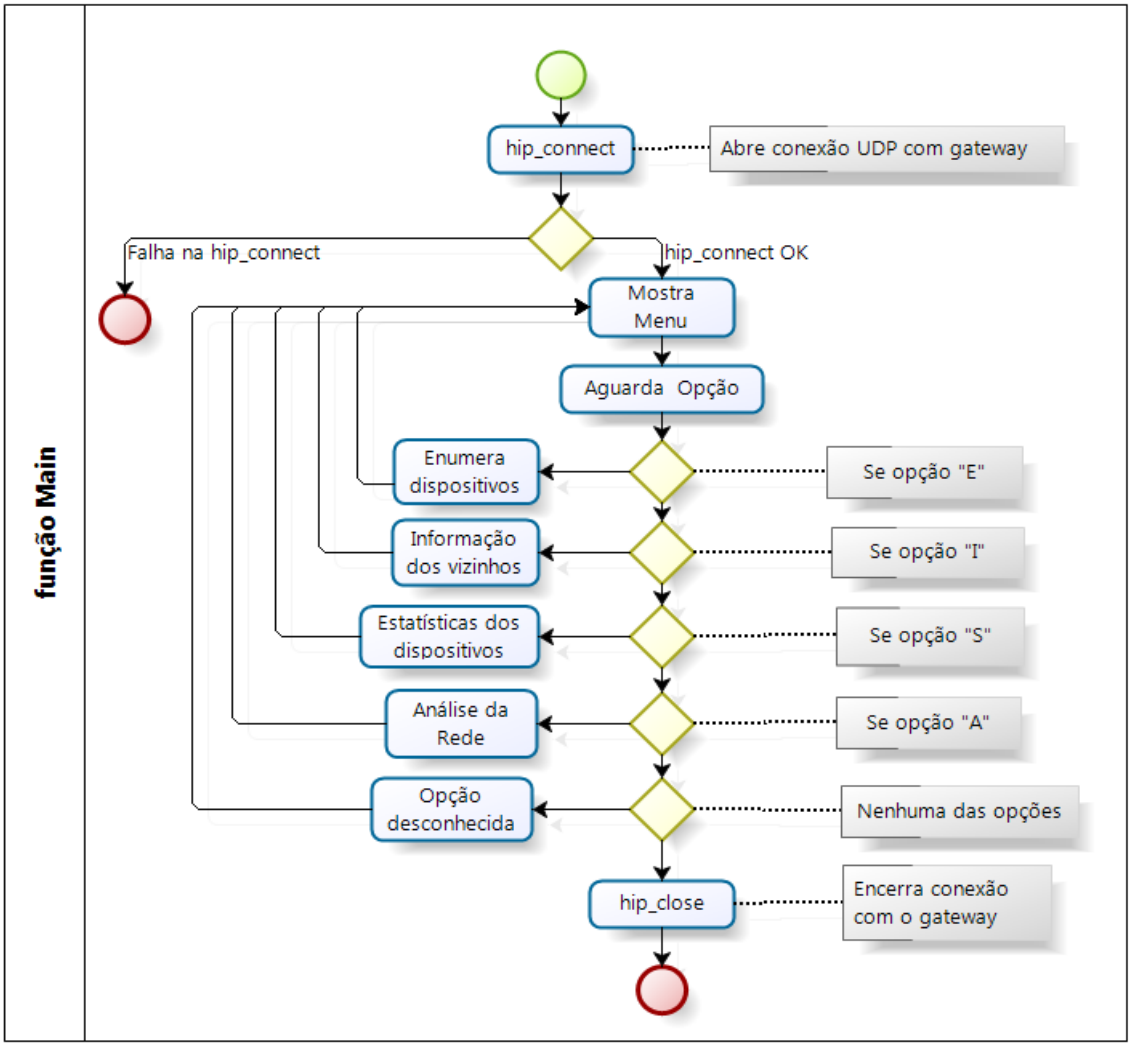

Figura 31. Fluxograma do Menu Principal.

Para cada uma das opções é atribuído uma função principal responsável por montar o frame UDP de acordo com os argumentos dos comandos utilizados. Cada comando possui

argumentos e tamanhos diferentes. Abaixo uma breve descrição de alguns detalhes relevantes sobre as funções utilizadas.

Hip_enum_nodes: esta função é relacionada ao comando 814. Após o delimitador HART são utilizados os seguintes argumentos 0x1F, 06, 0x03, 0x2E, 0x00, 16, 0, 0. Onde 0x1F refere-se ao comando estendido, no caso 31 em decimal o número 06 representa o número de bytes a partir deste parâmetro. Os valores 0x03 e 0x2E são o próprio comando 814. Estes parâmetros podem ser melhor entendidos verificando o capítulo de HART sobre UDP.

Hip_get_nickname: esta função obtém o endereço curto do dispositivo (2 bytes) identificado na especificação HART como apelido do dispositivo. O número atribuído a esta função é o 781. Sendo que os argumentos que variam no frame UDP são 0x1f, 2, 0x3, 0x0d. Como explicado anteriormente o primeiro valor é o comando estendido 31, número de bytes e o próprio comando respectivamente;

Hip_get_neighbor_health_list: esta função está associada ao comando 780 e também é responsável por montar os frames de requisição de acordo com os parâmetros necessários; Os parametros variáveis são: 0x1F, 4, 0x3, 0x0C, 0 e 99;

Hip_get_network_statistics: Esta função monta o frame para o comando 840, sendo que o comando 840 é um comando para o gateway. Há uma diferença que este comando não precisa variar os endereços dos dispositivos, logo que a requisição é enviada diretamente para o gateway. Este comando não utiliza nenhum argumento como visto a seguir 0x1F, 7, 0x3, 0x48

Hip_get_route_list: Montagem do frame associado ao comando 802. Os argumentos são, após a identificação do comando (0x03, 0x22) o índice da tabela de rotas no caso 0 e o número de entradas para leitura, 99. Na seqüência os valores são 0x1F, 4, 0x3, 0x22, 0, 99;

Hip_get_session_list: Associado ao comando 782. Os parâmetros são 0x1F, 4, 0x3, 0x0E, 0 e 99.

Hip_get_read_list: Responsável pelo frame do comando 784. Com os seguintes valores na entrada da função 0x1F, 4, 0x3, 0x0E, 0 e 99

Hip_get_service_list: Monta os frames relacionados ao comando 800, lista de serviços. Os valores utilizados são 0x1F, 4, 0x3, 0x20, 0 e 99;

Hip_get_*superframe*_list: Esta estrutura acompanha os dados referente ao comando 783 do WirelessHART sendo os parâmetros necessários igual 0x1F, 4, 0x3, 0x0F, 0 e 99.

Para todas estas funções os valores retornados apresentam tamanhos de frame diferentes e são montados em estruturas especificas para cada requisição. A Figura 32 representa o fluxograma básico destas funções que são utilizadas dentro das opções do aplicativo.

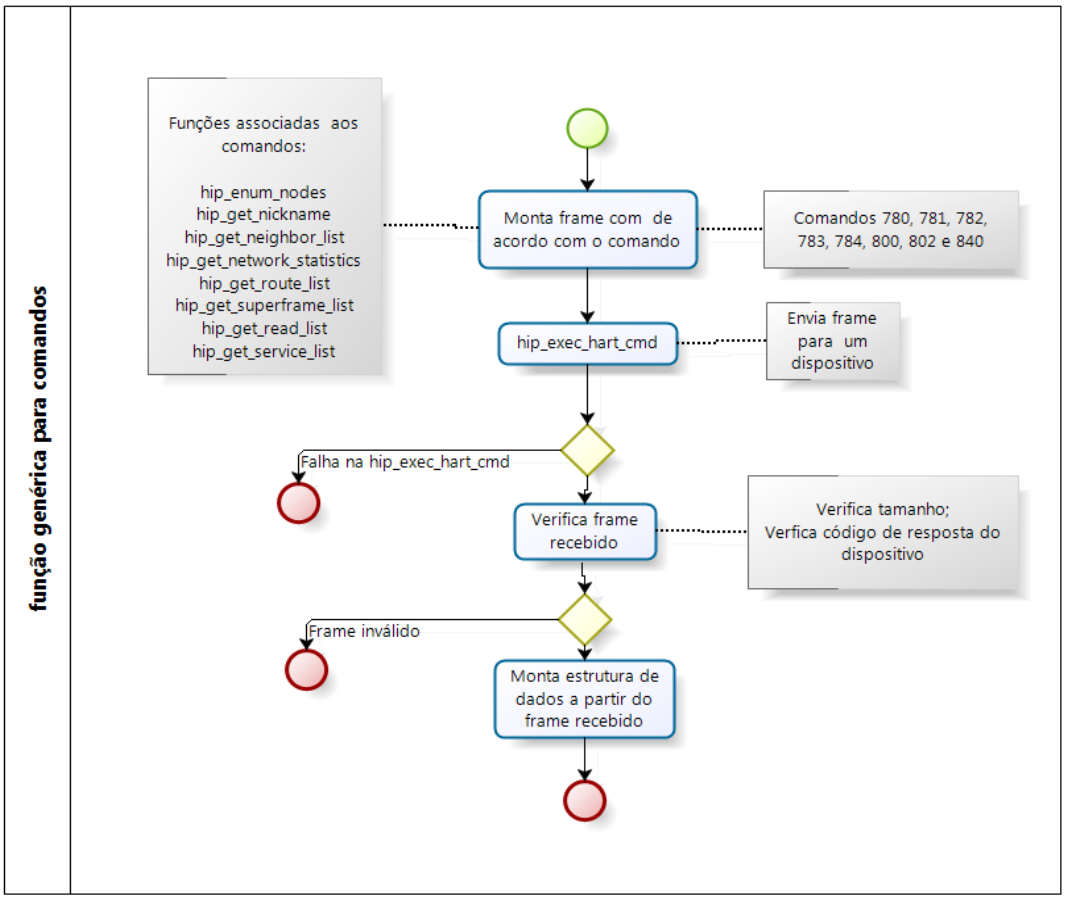

Figura 32. Fluxograma Genérico das Funções dos Comandos.

A função hip_exe_hart_cmd é a responsável pelo encapsulamento dos comandos HART sobre UDP também é a responsável por realizar as verificações necessárias quanto a possíveis erros. O fluxograma está demonstrado na Figura 33.

Devido ao fato de cada comando ter um número variável de bytes de resposta em função do número de dispositivos na rede que pode ser dinâmico cada comando possui a verificação do tamanho do frame recebido em função do menor número de bytes respondido independente do número de dispositivos. Caso a resposta não atenda a requisição é dado um erro e o software interroga o próximo dispositivo da lista. Ou se no caso de atraso de resposta, dependendo do código de resposta, o software aguarda e tenta aquisição novamente. Para códigos de resposta de valores 33 ou 34 a função deve enviar novamente o comando.

Tabela 13: Códigos do Mecanismo do Atraso de Respostas.

Mnemônico	Valor	Descrição
BUSY	32	Indica se o dispositivo está executando alguma operação.
DR_INITIATE	33	Indica o início de um DR. O dispositivo escravo precisa de mais tempo.
DR_RUNNING	34	Indica que o dispositivo ainda está processando o comando
DR_DEAD	35	Indica que nenhuma resposta foi recebida do dispositivo escravo.
DR_CONFLICT	36	Indica que o comando não pode ser processado.

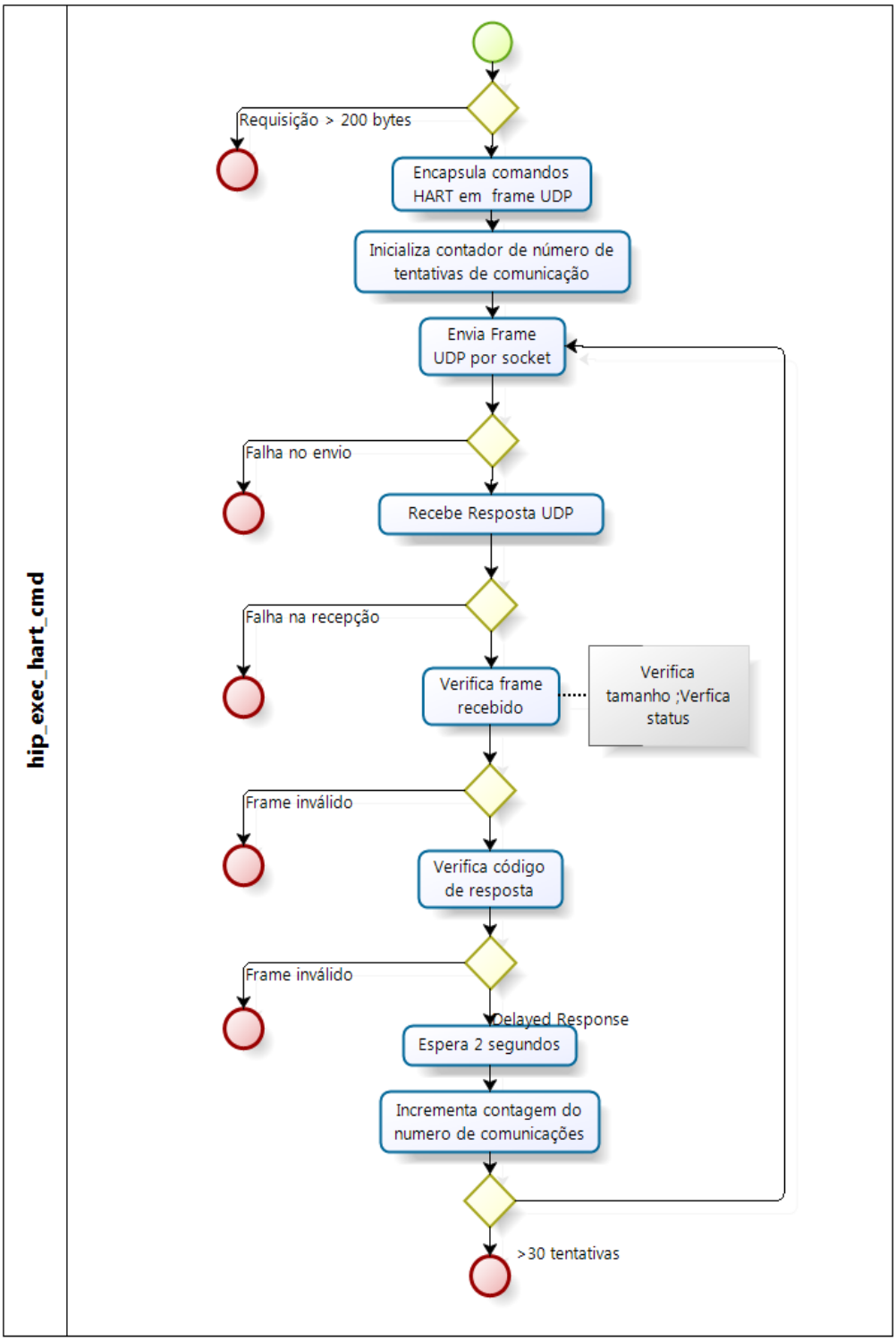

Figura 33. Fluxograma da Função Executa Comando.

Na Figura 34 está a opção que lista e imprime na tela os dispositivos que estão associados a rede WirelessHART. Nesta opção acontece a chamada da função hip_enum_nodes que está associada ao comando 814 do padrão WirelessHART.

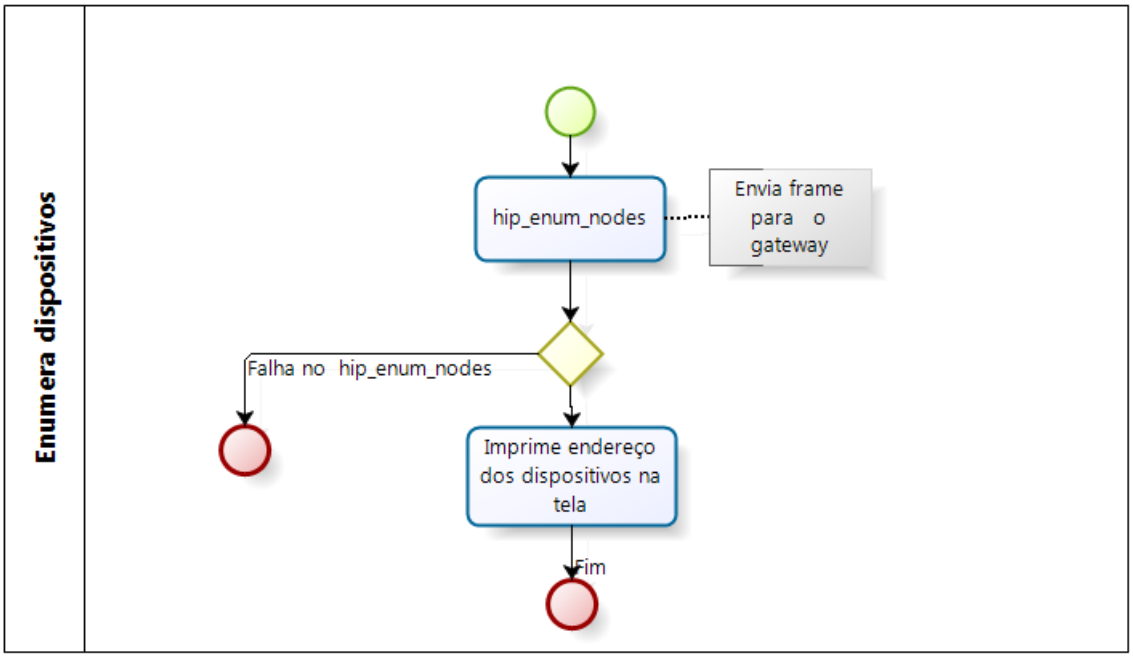

Figura 34. Fluxograma Opção de Listagem de Dispositivos.

O próximo fluxograma a seguir representa a opção de análise dos vizinhos e tem como saída quais dispositivos cada dispositivo tem ou pode estabelecer comunicação, o nível de sinal RSL entre eles, quantidade de pacotes transmitidos e recebidos. Com estes dados é possível realizar uma determinação prévia de quais são as possíveis rotas estabelecidas entre os dispositivos pelo gerenciador de rede, assim como através do nível de sinal RSL determinar a ordem de associação do dispositivo.

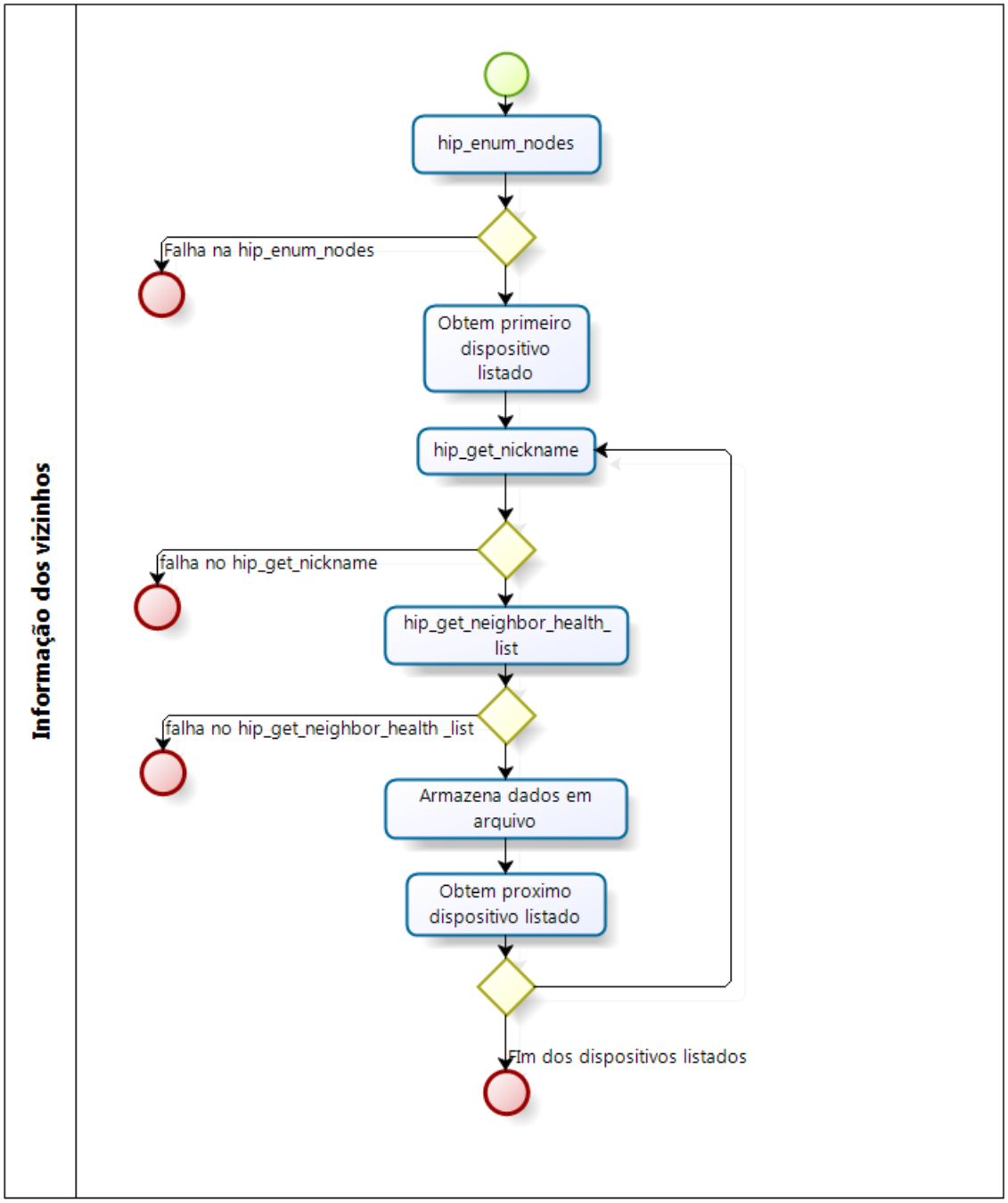

Figura 35. Fluxograma opção Dados dos Dispositivos.

A opção estatística dos dispositivos, ver Figura 36, é baseada no comando 840 do protocolo do WirelessHART e apresenta dados relevantes para a análise do comportamento da rede. O gerenciador de rede utiliza dados da mesma categoria para criar e modificar a topologia da rede. Este comando foi implementado e executa sempre junto com a pilha da

próxima opção do software (Análise da Rede) além de uma opção de acesso direto a estes dados.

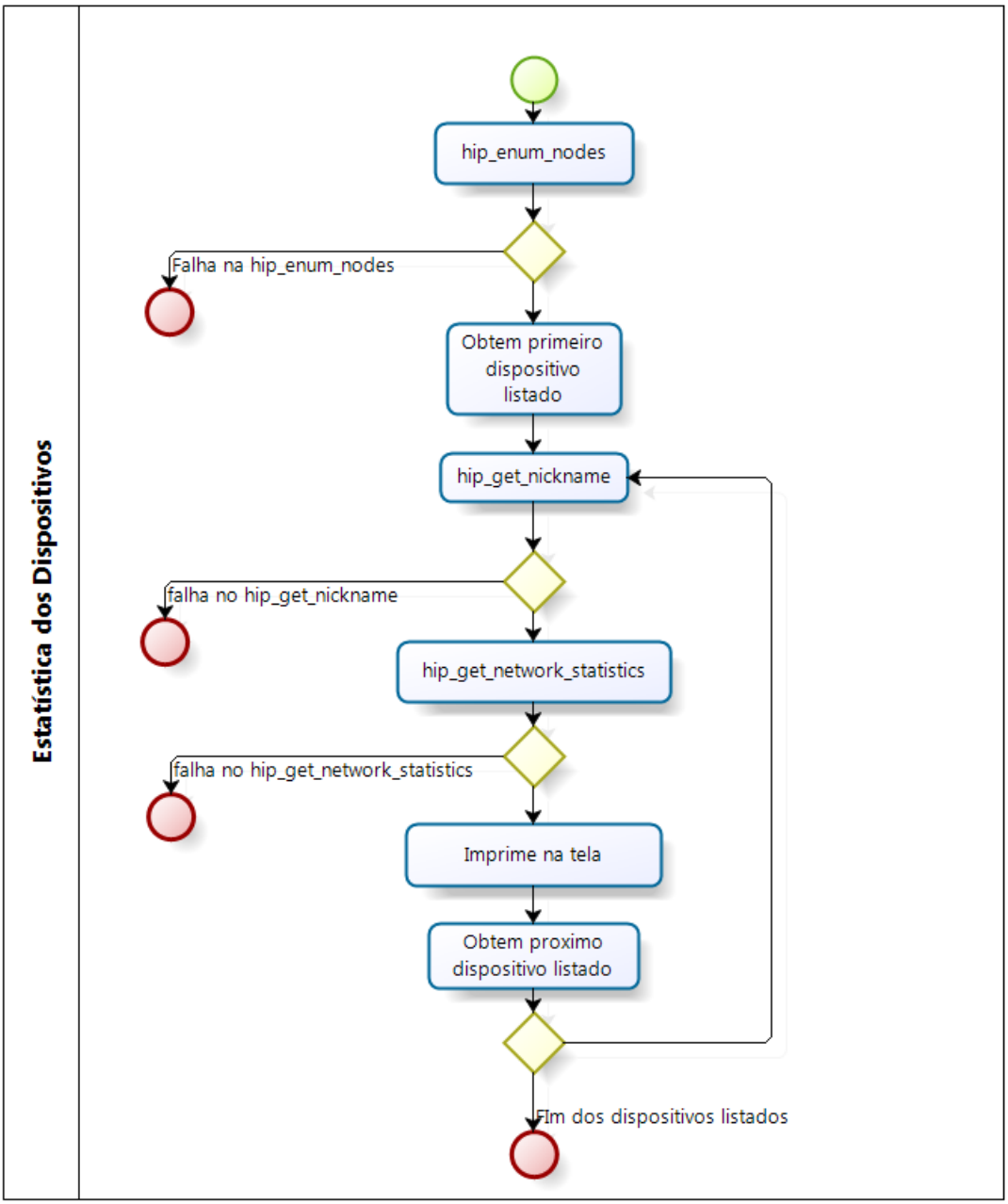

Figura 36. Fluxograma opção Dados Estatísticos dos Equipamentos.

A última opção do aplicativo é uma rotina que faz a solicitação de diversos dados para os dispositivos, sendo que primeiramente é solicitada uma identificação dos dispositivos que estão associados à rede (comando 814) e em seguida os dispositivos um a um são interrogados com os comandos, ver Figura 37. Esta opção faz uso de todos os comandos

WirelessHART implementados neste trabalho. Alguns dados são impressos na tela do usuário da aplicação, mas essencialmente esta opção registra os dados coletados em um arquivo do tipo csv (*comma separeted values*), para posterior análise.

Esta opção como visto na Figura 37 possui fora do laço principal a função hip_enum_nodes que tem como principal função verificar se algum novo dispositivo associou-se à rede. Após esta chamada os dispositivos são novamente interrogados e assim os dados continuam sendo registrados.

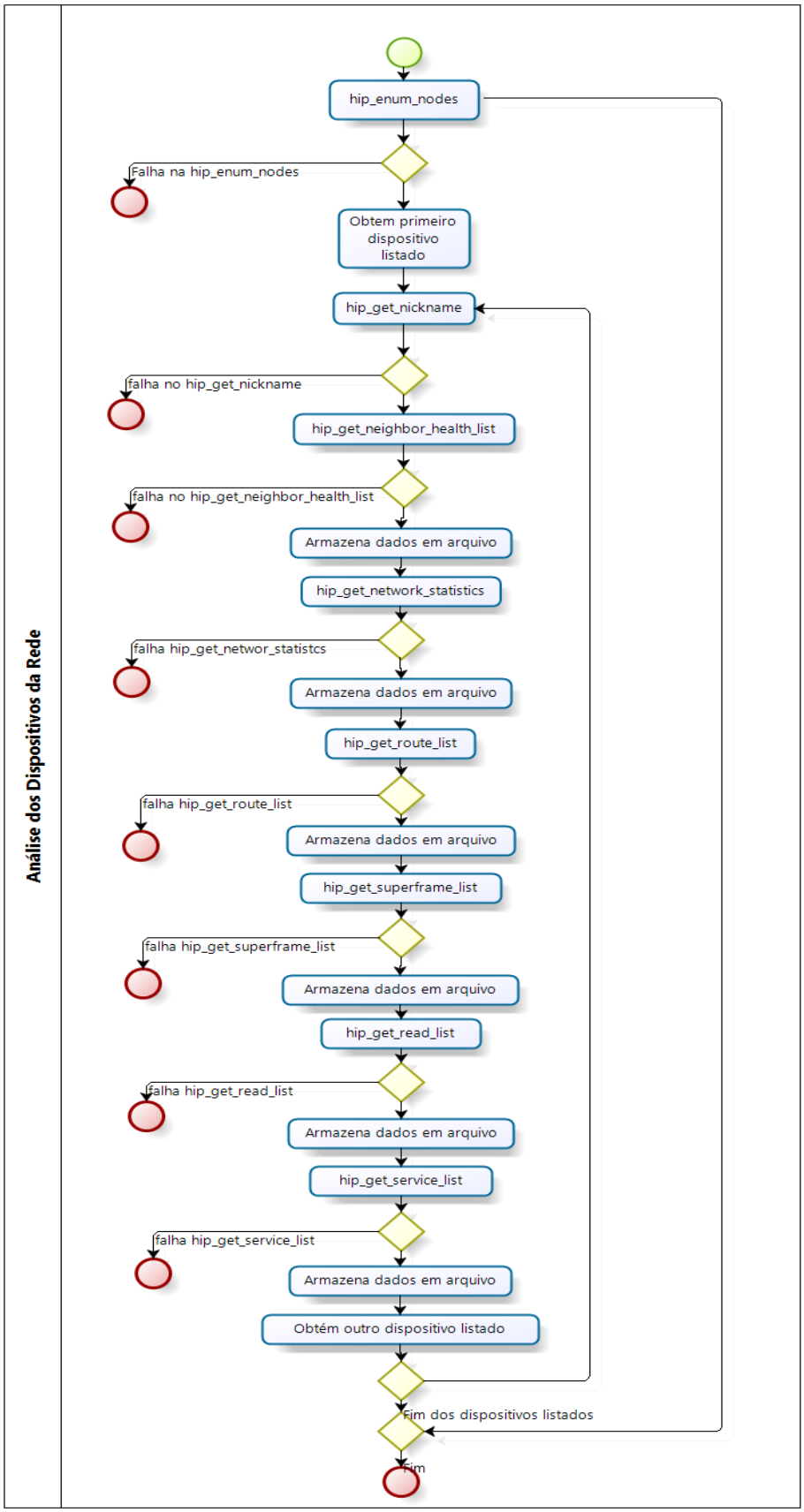

Figura 37. Fluxograma opção de Análise de Dispositivos da Rede.

8.4 ENSAIO REALIZADO

Para a coleta de dados através do software desenvolvido foram utilizados 2 equipamentos transmissores da Emerson TT 648, 3 dispositivos com modulo de comunicação 802.15.4 da Freescale e 1 rádio protótipo para uso com o protocolo WirelessHART além do Gateway 1420A da Emerson. Abaixo as principais características dos equipamentos utilizados:

1) Gateway:

O gateway interliga ao nível das camadas superiores, redes de protocolos diferentes. Por isso pode ser usada para interligar duas arquiteturas de comunicações completamente diferentes [15].

- Fabricante: Rosemount – Emerson Process Management;
- Modelo: 1420A
- Tensão de entrada: 24 VDC;
- Corrente necessária: 500 mA;
- Antena: Omnidirecional integrada (PBT/PC):
- Rádio freqüência 902-928 MHz (FHSS); 2.4-2.5 GHz (DSSS)
- RS-485: 2 *links* de comunicação para conexão Modbus
- Baud rate: 57600, 38400, 19200 ou 9600;
- Protocolo: Modbus RTU;
- Ethernet: porta de comunicação 10baseT/ 100base-TX;
- Suporte a Modbus TCP/IP e OPC com 32 bits;
- 100 dispositivos WirelessHART: taxa máxima de transmissão 60 s;
- 50 dispositivos WirelessHART: taxa máxima de transmissão 15 s;
- Latência da Rede:
- 100 dispositivos WirelessHART: 10 s;
- 50 dispositivos WirelessHART: 5 s;

2) Sensor Transmissor de temperatura[16]:

- Fabricante: Rosemount – Emerson Process Management;
- Modelo: 648;
- Alimentação: Bateria de lítio e cloreto de thionyl, substituível, não recarregável. Vida de 8 anos na taxa de uma atualização por minuto;
- Antena omnidirecional integrada (PTB/PC);

Figura 38. Sensor de Temperatura 648 - Emerson.

3) Rádio protótipo:

Módulo de comunicação para válvulas possui um firmware desenvolvido para o padrão WirelessHART [16].

- Principais características
- MCU 13224 da Freescale;
- Antena 20 dBM (100mW);
- Amplificador (LNA/PA CC2591);
- Conexão de programação e debug JTAG.

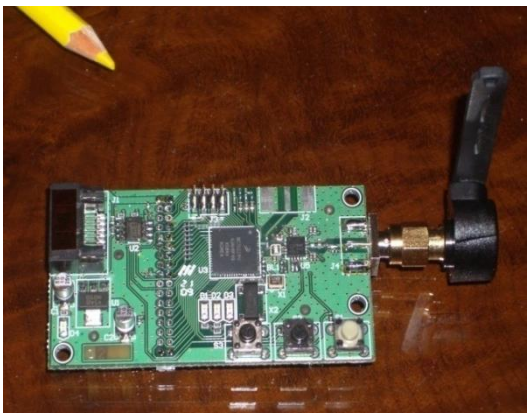

Figura 39. Rádio Protótipo.

4) Kit Freescale MC1322x:

Kit de desenvolvimento que opera no padrão IEEE 802.15.4, possui um microcontrolador da família ARM[17]. Neste dispositivo foi implementado um *stack* WirelessHART através do IDE (Integrated Development Environment) da IAR Systems.
Principais caracteristicas:

- Módulo de rádio de acordo com padrão 802.15.4
- Banda de operação 2.4 GHz ISM;
- Potência de saída -30dBm para 4dBm;
- Sensibilidade do receptor:
  <-96 dBm usando modo DCD ;
  <-100 dBm usando modo NCD;
- Sem componentes externos de RF;
- Componentes RF integrado e balun encapsulado;
- Porta de depuração JTAG;
- Processador 32bit ARM7;
- Cristal oscilador de 32,768 kHz;
- Tensão de operação: 2 a 3,6 V;
- Faixa de temperatura: -40ºC a +105ºC;

Figura 40. Módulo de rádio utilizado nos ensaios.

Os ensaios foram realizados nas dependências da Escola de Engenharia Elétrica da Universidade Federal do Rio Grande do Sul, abrangendo os espaços do laboratório LASCAR e o setor de entrada da escola, os testes foram realizados nestes locais com o objetivo de garantir um espaçamento adequado entre os dispositivos e simular comunicação indireta de alguns módulos com o gateway sendo possível avaliar os dados registrados e identificar os

possíveis caminhos determinados pelo gerenciador de rede. Na Figura 41 uma representação da distribuição dos módulos assim como alguns dos obstáculos presentes no local de ensaio.

Inicialmente a rede operou com apenas 4 nodos (dispositivos de apelidos 2, 3 4, 5 e 6. Após um período de monitoração foi adicionado o dispositivo de apelido 7. O software após estabelecer conexão com o gateway passou a registrar os diversos dados associados aos comandos do protocolo WirelessHART.

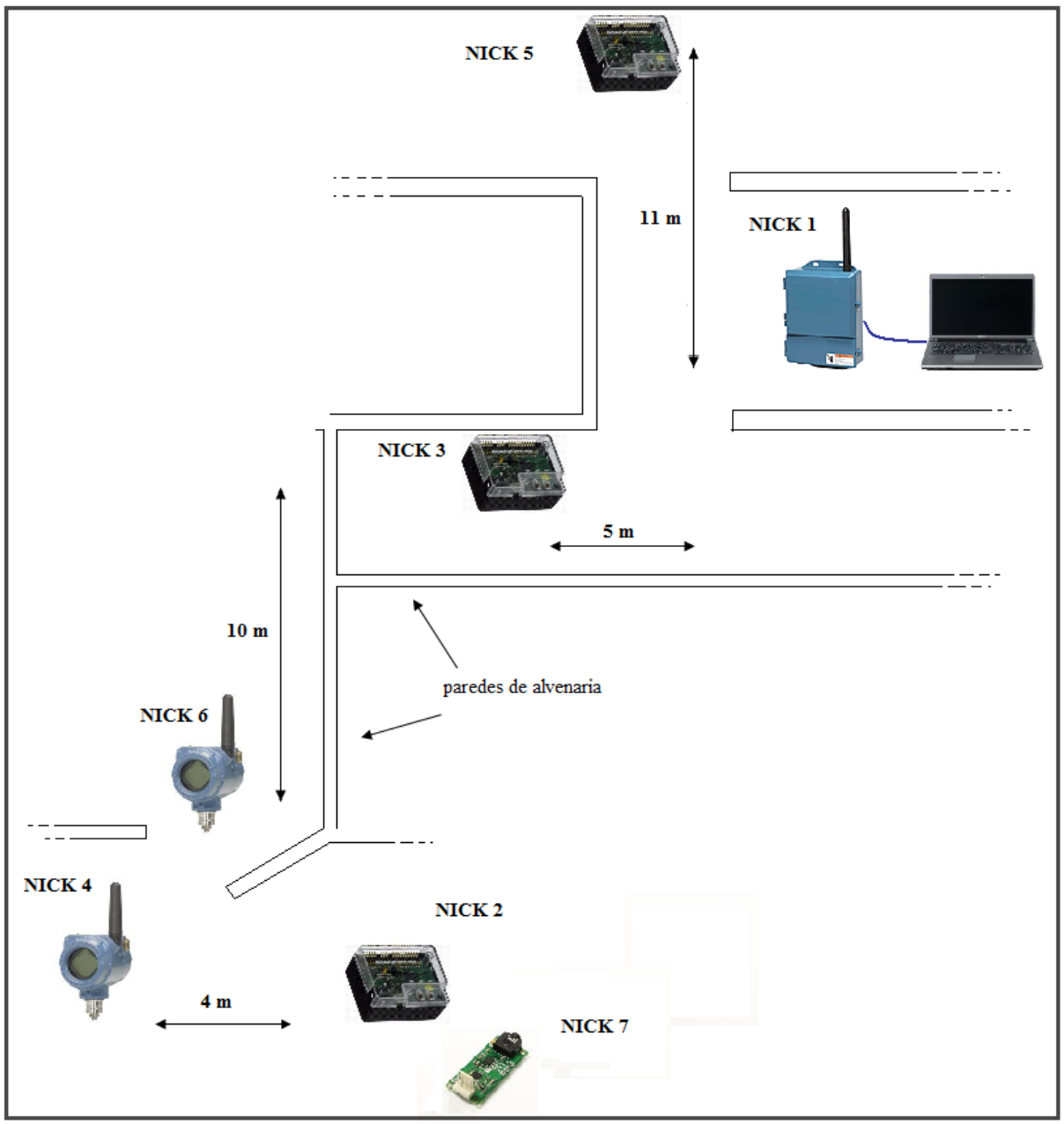

Figura 41. Distribuição dos Dispositivos para ensaio.

8.4.1 ANÁLISE DOS DADOS

O tempo e os canais de comunicação utilizados podem ser considerados como as duas primeiras dimensões do WirelessHART [3]. A terceira dimensão é o espaço (distância). Os dispositivos podem ser instalados em diversos locais em uma planta de processo. Uma característica importante como conseqüência desta distribuição no espaço é a qualidade do sinal de rádio freqüência. Um dispositivo apresenta uma lista de outros dispositivos o qual pode estabelecer comunicação, como a qualidade do sinal pode variar implicando na descoberta de novos vizinhos em potencial para estabelecer comunicação, perda de dispositivos vizinhos ou ainda como parâmetro para manter ou não um dispositivo na rede. Devido ao local do ensaio realizado ter sido na área da Escola entre os nodos houve movimentação de usuários diversos (estudantes, funcionários, professores, etc.) os quais possivelmente influenciaram na variação dos dados registrados. A duração do ensaio foi de quatro horas aproximadamente entre às 8 horas e 11 horas e 30 minutos. Entre os dados registrados pelo aplicativo é possível observar a sensibilidade do sinal para cada dispositivo, assim como identificar quais são os dispositivos vizinhos de cada elemento da rede. Neste primeiro conjunto de dados já é permitido limitar o número de possíveis dispositivos os quais podem estabelecer roteamento de mensagens.

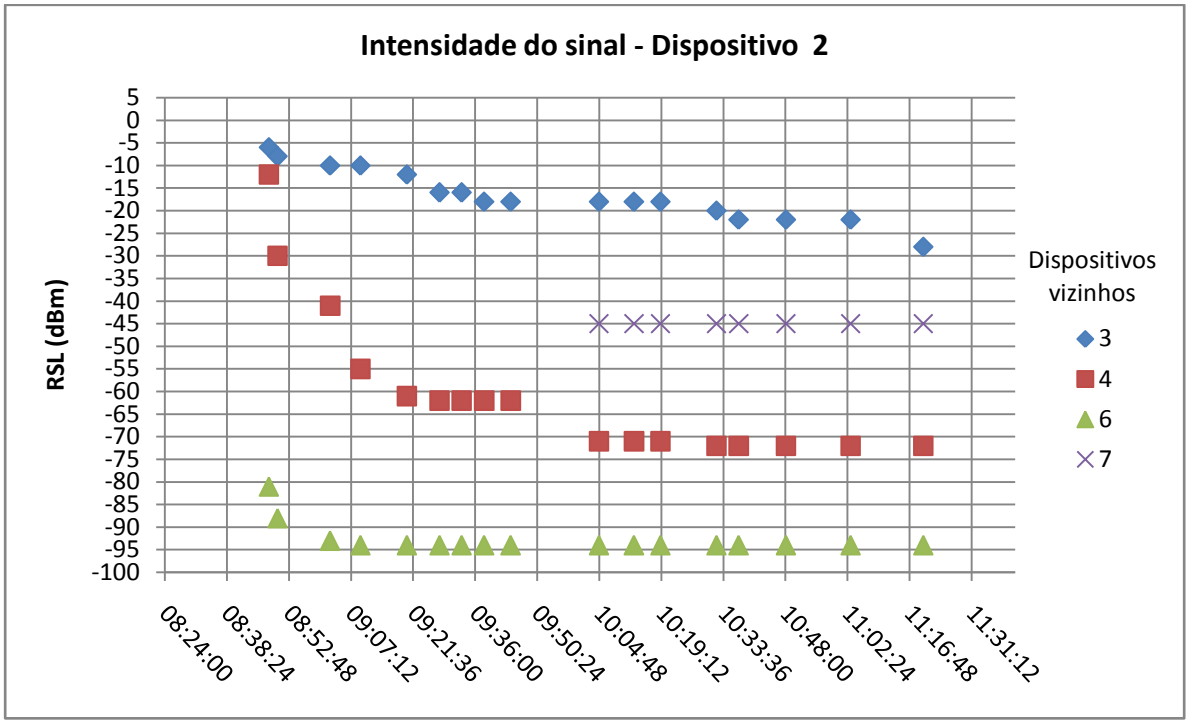

Figura 42. Relação de intensidade de Sinal do Dispositivo 2.

Na Figura 42, os dispositivos 6, 3 e 7 mantiveram uma relação de sinal razoável e coerente, um valor aproximadamente constante. O dispositivo 4 em relação ao dispositivo 2 apresentou uma variação grande, aproximadamente 60 dBm, na inicialização dos registros. É necessário na continuidade deste trabalho verificar as possíveis causas destes eventos.

No gráfico obtido a seguir, ver Figura 43, com exceção do dispositivo 5 e 6, foi apresentado uma relação de nível de sinal satisfatório. Este gráfico ainda mostra um evento de grande importância referente ao momento que o dispositivo 3 perdeu conexão com a rede, no caso entre o período das 10 às 11 horas, onde não foi registrado níveis de sinal para nenhum dispositivo vizinho.

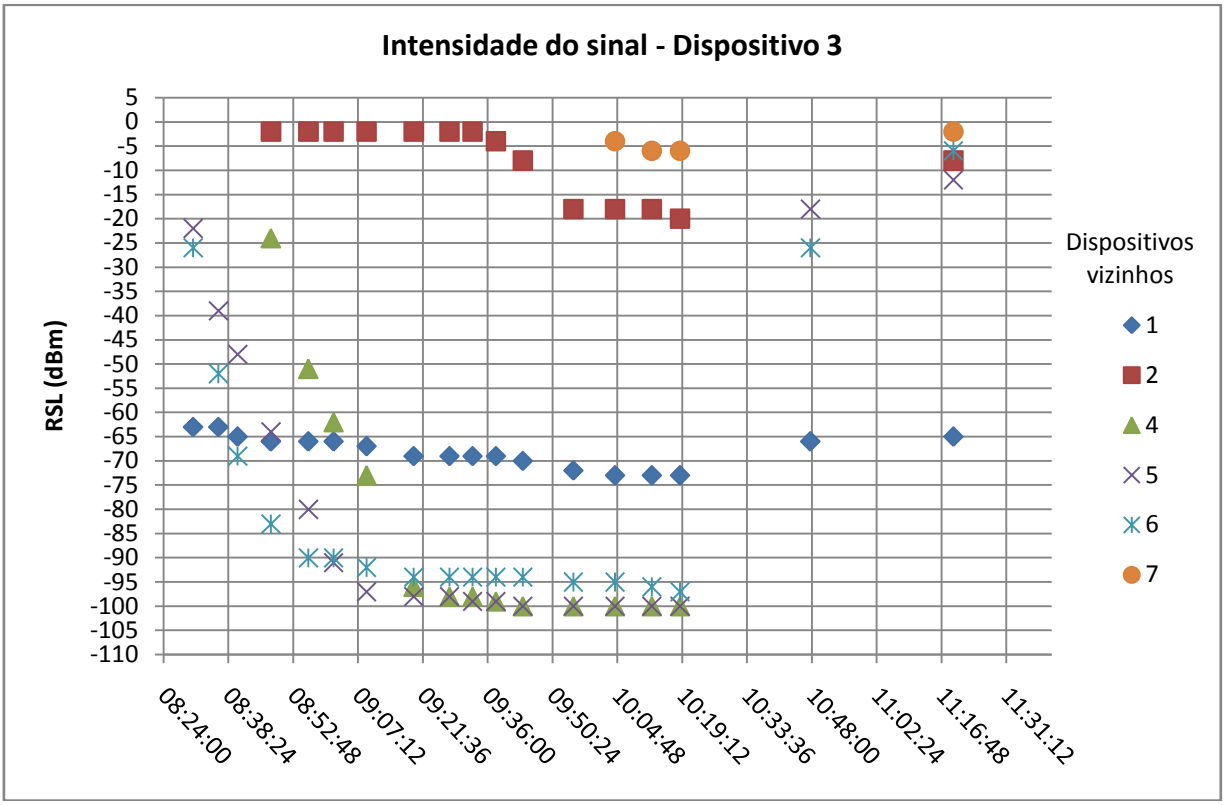

Figura 43. Relação de intensidade do Sinal do Dispositivo 3.

O dispositivo 4 não tem comunicação direta com o gateway (dispositivo 1). Nesse gráfico, ver Figura 44, pode-se verificar o ótimo nível de sinal que é mantido com o dispositivo 6. Os dispositivos 4 e 6 são produtos comerciais enquanto os outros dispositivos utilizados ainda estão em fase de desenvolvimento.

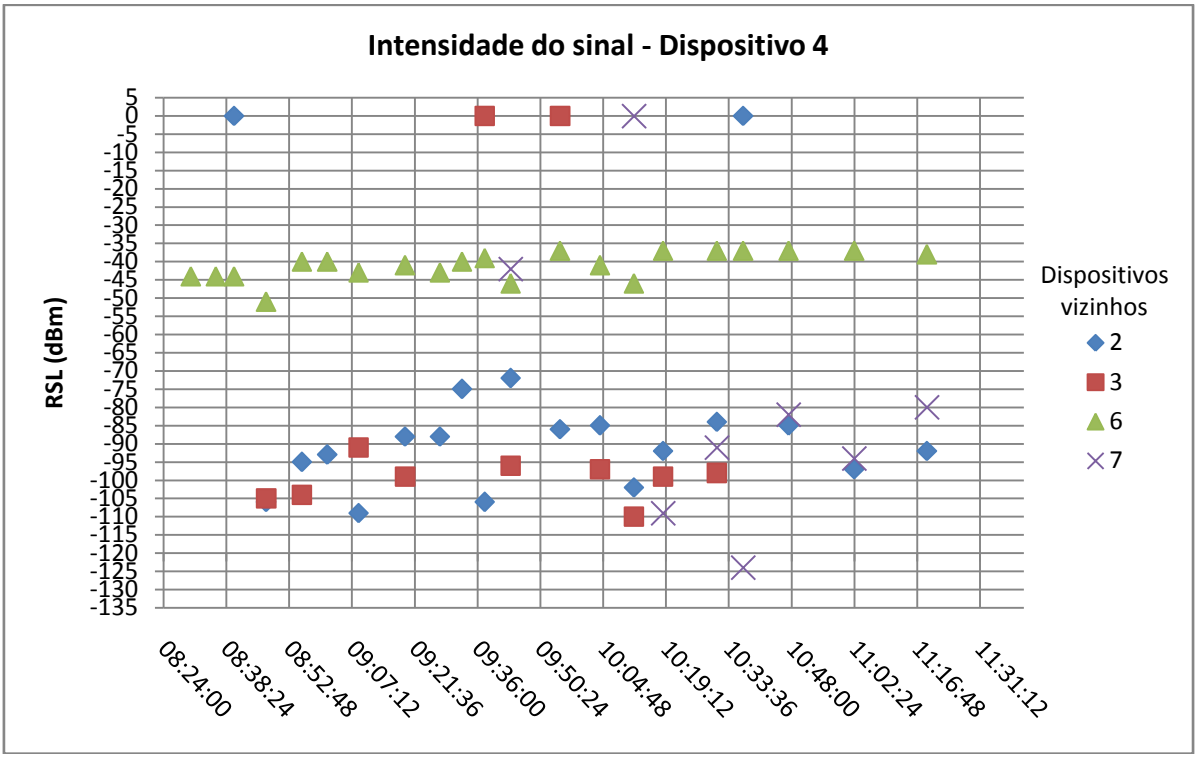

Figura 44. Relação de intensidade de Sinal do Dispositivo 4.

Na Figura 45 fica claro que o dispositivo 5 também perdeu conexão com a rede no intervalo de 10:19 até 11 horas aproximadamente.

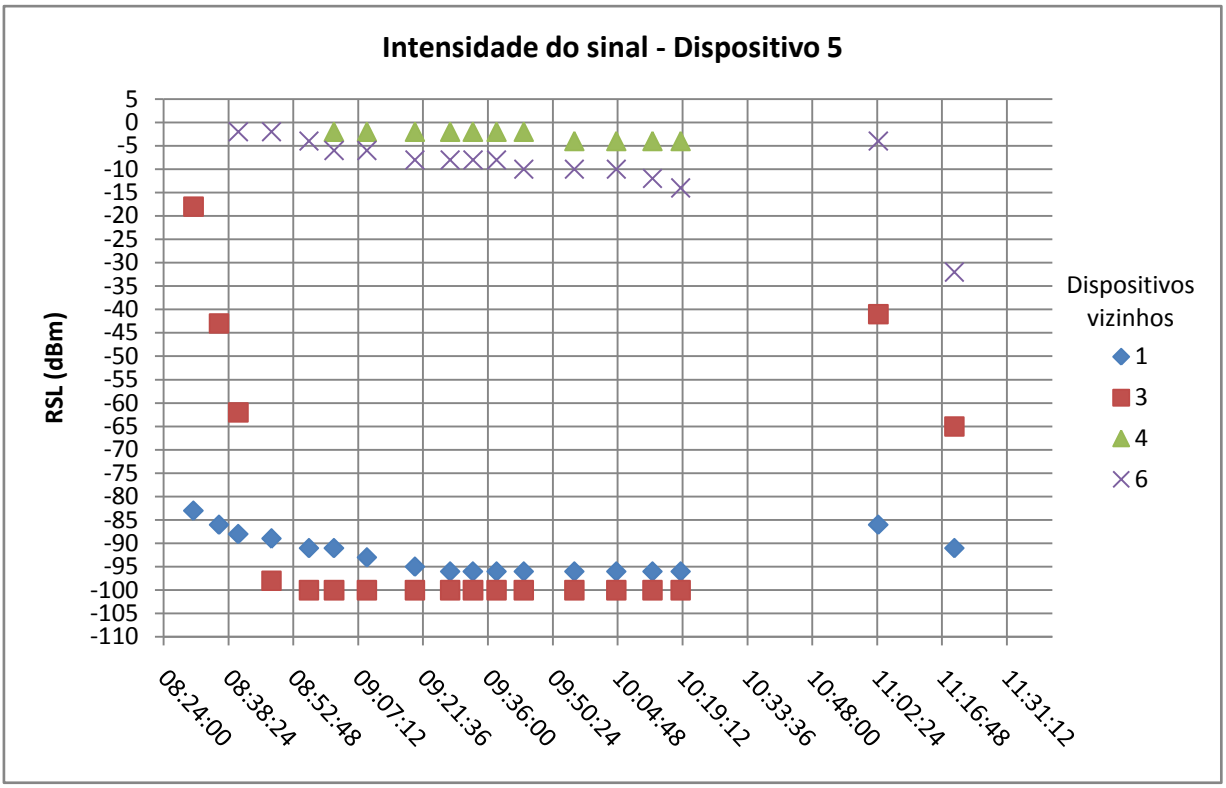

Figura 45. Relação de intensidade de Sinal Dispositivo 5.

Na Figura 46, todas as intensidades medidas mostraram-se com uma pequena variação. Destaque para a relação de sinal entre os dispositivos 6 e 4, os quais, encontram-se com linha de visada na localização física do cenário de ensaio além da curta distância entre eles (aproximadamente 7 metros). Esse comportamento foi esperado entre tais dispositivos.

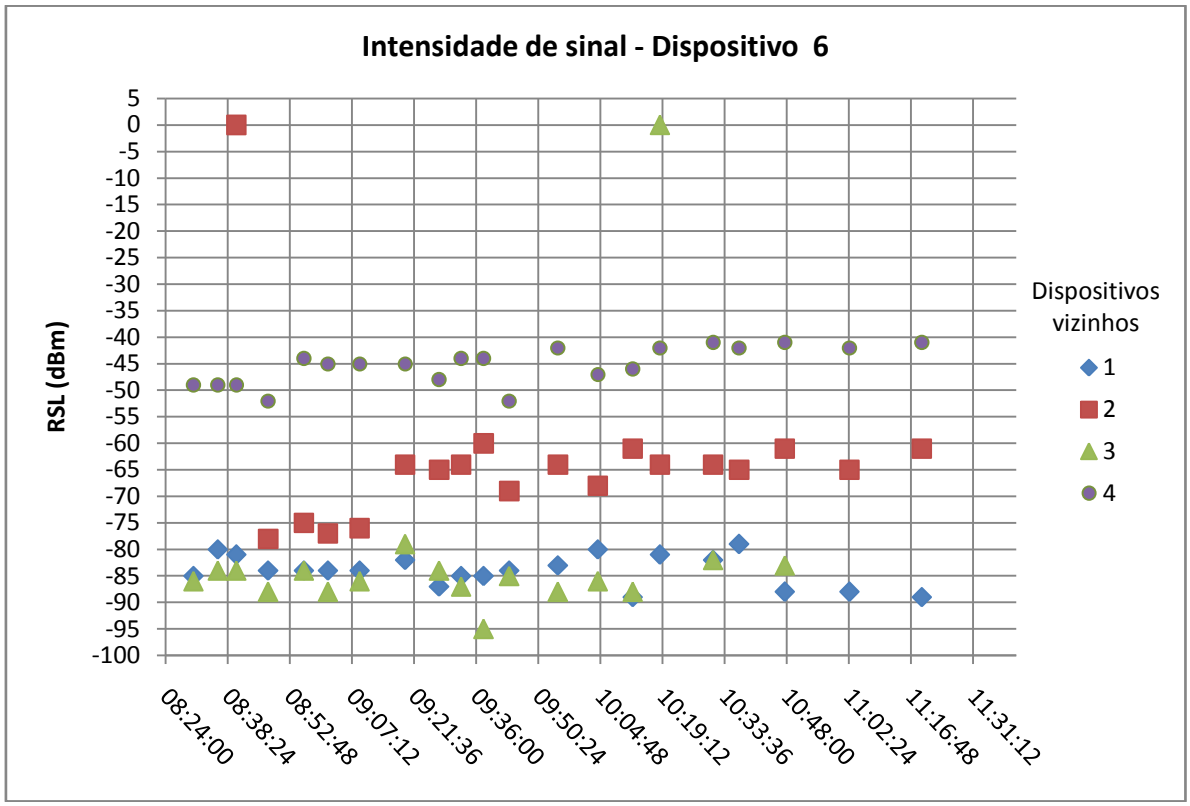

Figura 46. Relação de intensidade de Sinal do Dispositivo 6.

O dispositivo 7 foi acionado e integrado na rede durante o processo de monitoramento das variáveis, logo é possível apenas verificar sua entrada na rede às 10 horas aproximadamente e a partir deste momento avaliar seu nível de sinal em relação aos seus vizinhos, ver Figura 47.

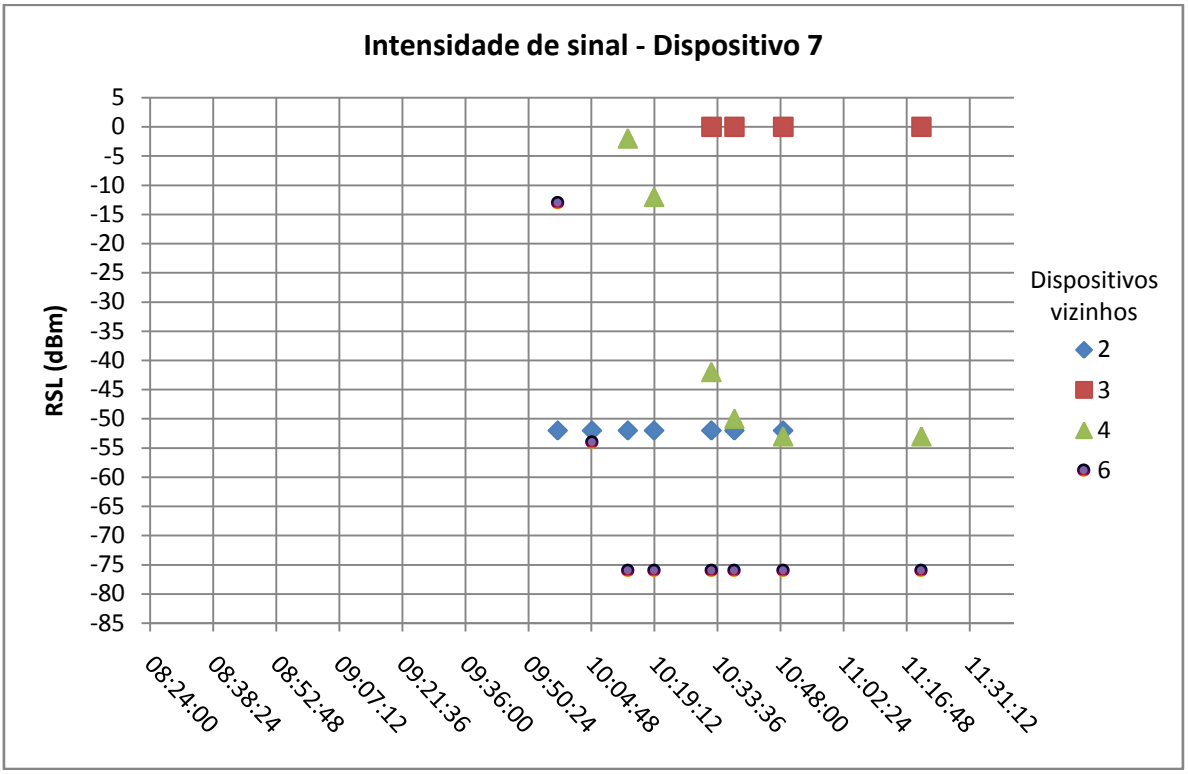

Figura 47. Relação de intensidade de Sinal do Dispositivo 7.

Na análise dos dados obtidos nesse ensaio foi possível pré determinar possíveis rotas de pacotes entre os dispositivos assim como :

Dispositivo 2: Nós 3, 4, 6 e 7 como vizinhos;

Dispositivo 3: Nós 1, 2, 4, 5, 6 e 7 como vizinhos;

Dispositivo 4: Nós 2, 3, 6 e 7 como vizinhos;

Dispositivo 5: Nós 1, 3, 4 e 6 como vizinhos;

Dispositivo 6: Nós 1, 2, 3 e 4 como vizinhos;

Dispositivo 7: Nós 2, 3, 4 e 6 como vizinhos.

Na Figura 48 está a representação dos dispositivos vizinhos do gateway e, como exemplo, os vizinhos que o dispositivo 2 consegue “escutar”.

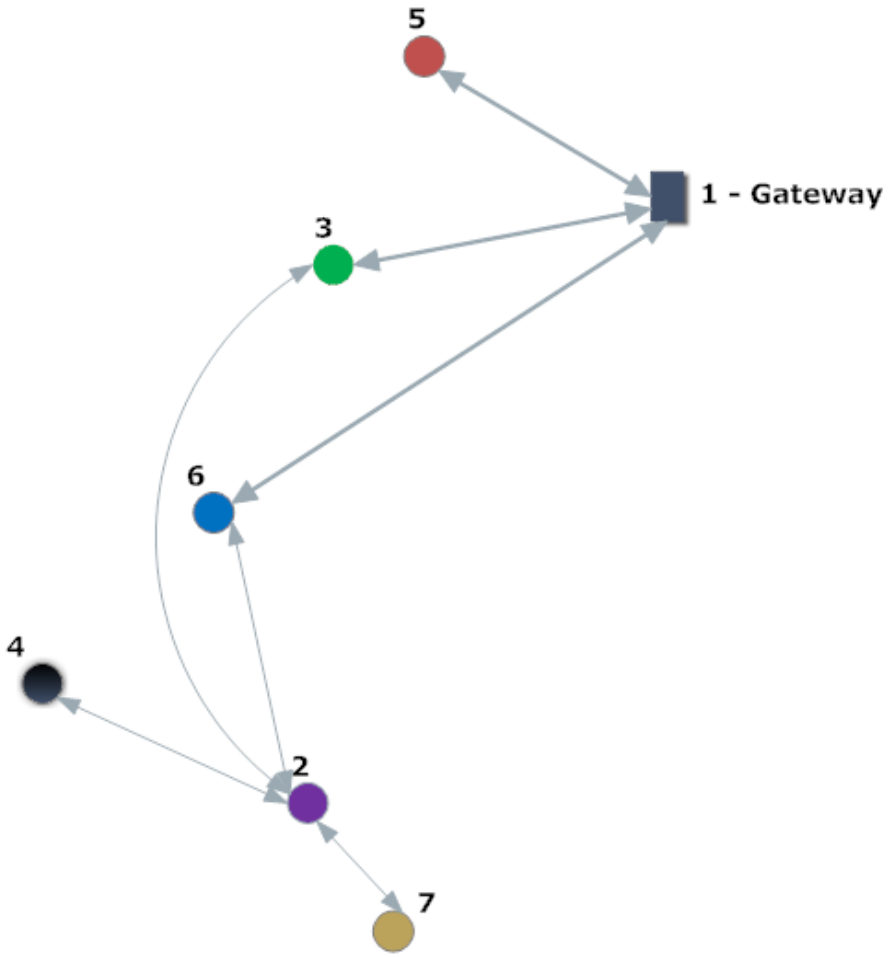

Figura 48. Vizinhos do Gateway e vizinhos do Dispositivo 2.

No próximo registro de dados através do comando 783, *Read Superframe List*, foi obtido dados sobre as propriedades dos *superframe*s ativos na rede. O *superframe* é um conjunto de *slots* sendo que cada *slot* possui um *link* associado a ele. O aplicativo registra os dados refente aos *superframes* ativos na rede, pode-se verificar neste ensaio três *superframes* ativos e suas propriedades. A tabela 13 apresenta um intervalo dos registros. Estes valores não tiveram variação para seus respectivos dispositivos com exceção de quando algum dispositivo desconectou-se da rede.

Tabela 14: Lista de *Superframes.*

Date	Hour	Nickname	Number Active *Superframes*	*Superframe* ID	Number of *slots* (in this *superframe*)	Flag *Superframe*
30/11/2010	09:59:21	4	3	4	128	1
30/11/2010	09:59:21	4	3	1	256	1
30/11/2010	09:59:21	4	3	0	1024	1
30/11/2010	09:59:59	6	3	4	128	1
30/11/2010	09:59:59	6	3	1	256	1
30/11/2010	09:59:59	6	3	0	1024	1
30/11/2010	10:00:05	3	3	1	256	1
30/11/2010	10:00:05	3	3	0	1024	1
30/11/2010	10:00:05	3	3	4	128	1
30/11/2010	10:00:17	5	3	1	256	1
30/11/2010	10:00:17	5	3	0	1024	1
30/11/2010	10:00:17	5	3	4	128	1
30/11/2010	10:00:35	2	3	1	256	1
30/11/2010	10:00:35	2	3	0	1024	1
30/11/2010	10:00:35	2	3	4	128	1
30/11/2010	10:00:57	7	3	1	256	1
30/11/2010	10:00:57	7	3	0	1024	1
30/11/2010	10:00:57	7	3	4	128	1

A seguir foram obtidas informações relacionadas aos *links* dos dispositivos. O *link* representa os parâmetros necessários para mover um pacote em um salto (entre nodos adjacentes). Os *links* estão associados com um dispositivo especifico, para cada *link* há *slots* alocados para um ou mais dispositivos. Os *links* estão endereçados pela sua posição na lista de *link* dos dispositivos e não possuem uma implementação particular. Algumas propriedades dos *links*:

- Vizinho: Apelido do vizinho associado para este *link* ou 0xFFFF se for broadcast;
- Opções do *link*: Cada *link* possui exclusivamente um *slot* associado a um *superframe* com as opções *(transmit, receive, shared*) e os tipos;
- Tipo do *link*:
    - Normal: *Link* para troca de dados O *link* normal tem um endereço de origem e um endereço de destino. A fonte usa esta ligação para transmitir uma mensagem ao destino;
    - Broadcast: *Link* para anúncio da rede. Este *link* está associado a um dispositivo, o qual é o remetente da ligação. O remetente envia mensagens de difusão não reconhecidas (sem ACK), o endereço de destino é 0xFFFF;

- Discovery: *Link* utilizado para manter a conectividade entre os dispositivos. Tem como principal função permitir que outros dispositivos descubram novos vizinhos;
- Join: *Link* de associação. Também está associado a um dispositivo que pode ser o destino ou a fonte, contém a informação necessária para associação de um novo dispositivo.

Segue síntese dos dados obtidos de cada dispositivo referente às propriedades dos *links*, nestas tabelas estão registrados principais links estabelecidos entre os dispositivos:

Tabela 15: *Links* do Dispositivo 2.

Vizinho	**Inicio**	**Fim**	***Superframe* ID**	**Número do *Slot***	**Opção do *Link***	**Tipo do *Link***
6	08:44:21	11:12:48	0	605	Transmit	Normal
6	08:44:21	11:12:48	0	1	Transmit|Receive	Discovery
6	08:44:21	11:12:48	0	93	Transmit	Normal
6	08:44:21	11:12:48	0	349	Transmit	Normal
6	08:44:21	11:12:48	0	861	Transmit	Normal
6	08:44:21	11:12:48	0	29	Transmit	Normal
f980	08:44:21	11:12:48	1	74	Transmit	Join
f980	08:44:21	11:12:48	0	669	Receive	Join
Ffff	08:44:21	11:12:48	1	54	Receive	Broadcast
Ffff	08:44:21	11:12:48	4	24	Transmit	Broadcast

Na primeira tabela pode-se notar que o dispositivo 2 possui *links* diretos com dispositivo 6, que por sua vez tem *links* com o gateway.

Tabela 16: *Links* do Dispositivo 3.

Vizinho	**Inicio**	**Fim**	***Superframe* ID**	**Número do *Slot***	**Opção do *Link***	**Tipo do *Link***
1	08:33:38	10:57:43	0	673	Transmit	Normal
1	08:33:38	10:57:43	0	1	Transmit|Receive	Discovery
1	08:33:38	10:57:43	0	161	Transmit	Normal
1	08:33:38	10:57:43	0	417	Transmit	Normal
1	08:33:38	10:57:43	0	930	Transmit	Normal
f980	08:33:38	10:57:43	1	46	Transmit	Join
f980	08:33:38	10:57:43	0	180	Receive	Join
Ffff	08:33:38	10:57:43	1	37	Receive	Broadcast
Ffff	08:33:38	10:57:43	1	52	Receive	Broadcast
Ffff	08:33:38	10:57:43	4	98	Transmit	Broadcast

Nos dados registrados o equipamento de apelido 3 fez transmissão com o gateway. E dispositivo de apelido 4 tem *links* de comunicação com diversos dispositivos, mas não com o gateway. Este dispositivo participou do roteamento de mensagens.

Tabela 17: *Links* do Dispositivo 4.

Vizinho	**Inicio**	**Fim**	***Superframe* ID**	**Número do *Slot***	**Opção do *Link***	**Tipo do *Link***
2	08:43:47	11:11:59	0	1007	Receive	Normal
3	08:43:47	10:21:56	0	948	Transmit	Normal
3	08:43:47	10:21:56	1	46	Receive	Broadcast
5	09:23:51	09:29:59	0	387	Transmit	Normal
5	09:23:51	09:29:59	1	58	Receive	Broadcast
6	08:33:10	11:11:59	0	222	Transmit	Normal
6	08:33:10	11:11:59	0	477	Transmit	Normal
6	08:33:10	11:11:59	0	733	Transmit	Normal
6	08:33:10	11:11:59	0	989	Transmit	Normal
6	08:33:10	11:11:59	1	54	Receive	Broadcast
7	09:41:19	11:11:59	0	751	Receive	Normal
7	09:41:19	11:11:59	0	239	Receive	Normal
7	09:41:19	11:11:59	0	496	Receive	Normal
7	09:41:19	11:11:59	0	751	Receive	Normal
7	09:41:19	11:11:59	0	751	Receive	Normal
ffff	08:33:10	11:11:59	0	1	Transmit|Receive	Discovery
ffff	08:33:10	11:11:59	0	495	Receive	Join
ffff	08:33:10	11:11:59	1	65	Transmit	Join
ffff	08:33:10	11:11:59	4	86	Transmit	Broadcast

Na Tabela 18 o dispositivo 5 obteve *links* de comunicação com o gateway e o gerenciador de redes (F980).

Tabela 18: *Links* do Dispositivo 5.

Vizinho	**Inicio**	**Fim**	***Superframe* ID**	**Número do *Slot***	**Opção do *Link***	**Tipo do *Link***
1	08:33:44	11:12:17	0	146	Transmit	Normal
1	08:33:44	11:12:17	0	1	Transmit|Receive	Discovery
1	08:33:44	11:12:17	0	402	Transmit	Normal
1	08:33:44	11:12:17	0	658	Transmit	Normal
1	08:33:44	11:12:17	0	914	Transmit	Normal
f980	08:33:44	11:12:17	1	58	Transmit	Join
f980	08:33:44	11:12:17	0	899	Receive	Join
ffff	08:33:44	11:12:17	1	37	Receive	Broadcast
ffff	08:33:44	11:12:17	4	4	Transmit	Broadcast
ffff	08:33:44	11:12:17	1	67	Receive	Broadcast

O dispositivo 6 serviu de ponte para comunicação de outros dispositivos com o gateway. É possível verificar na Tabela 18 que além do gateway obteve *links* de transmissão e recepção com nodos adjacentes. Ainda o *link* com o dispositivo 2 complementa os dados relacionados a tabela do dispositivo 2.

Tabela 19: *Links* do Dispositivo 6

Vizinho	**Inicio**	**Fim**	***Superframe* ID**	**Número do *Slot***	**Opção do *Link***	**Tipo do *Link***
1	08:33:32	11:12:13	0	98	Transmit	Normal
1	08:33:32	11:12:13	0	226	Transmit	Normal
1	08:33:32	11:12:13	0	577	Transmit	Normal
1	08:33:32	11:12:13	0	929	Transmit	Normal
2	08:43:53	11:12:13	0	29	Receive	Normal
2	08:43:53	11:12:13	0	61	Receive	Normal
2	08:43:53	11:12:13	0	93	Receive	Normal
2	08:43:53	11:12:13	0	124	Receive	Normal
3	08:33:32	08:38:13	0	692	Transmit	Normal
4	08:33:32	08:38:13	0	222	Receive	Normal
4	08:33:32	08:38:13	0	477	Receive	Normal
4	08:33:32	08:38:13	0	733	Receive	Normal
Ffff	08:33:32	11:12:13	0	1	Transmit|Receive	Discovery
Ffff	08:33:32	11:12:13	0	221	Receive	Join

O dispositivo 7 também não tem linha com o gateway e deve fazer uso de outros dispositivos intermediários para transmitir e receber pacotes. Seus nodos adjacentes são dispositivo 2 e 4, os quais estabelece *link* normal.

Tabela 20: *Links* do Dispositivo 7

Vizinho	**Inicio**	**Fim**	***Superframe* ID**	**Número do *Slot***	**Opção do *Link***	**Tipo do *Link***
2	10:01:46	10:58:56	0	157	Transmit	Normal
2	10:01:46	10:58:56	0	1	Transmit|Receive	Discovery
2	10:01:46	10:58:56	0	412	Transmit	Normal
2	10:01:46	10:58:56	0	668	Transmit	Normal
2	10:01:46	10:58:56	0	924	Transmit	Normal
2	10:01:46	10:58:56	0	28	Transmit	Normal
4	10:35:02	10:58:56	0	751	Transmit	Normal
f980	10:01:46	10:58:56	1	86	Transmit	Join
f980	10:01:46	10:58:56	0	445	Receive	Join
Ffff	10:01:46	11:58:56	1	74	Receive	Broadcast
Ffff	11:01:46	12:58:56	4	109	Transmit	Broadcast

Na Tabela 21, uma outra abordagem de visualização para mostrar o registro da comunicação entre os nodos 2 e 6. Sendo que o dispositivo de apelido 2 não tem comunicação direta com o gateway e utiliza o dispositivo 6 como nó intermediário para estabelecer a comunicação final com o gateway.

Tabela 21: *Links* registrado entre Dispositivos 2 e 6 .

Date	Hour	Nickname	*Superframe* ID	*Slot* number(*link*)	Channel *Offset*	Neighbor	*Link* Options	*Link* type
30/11/2010	08:43:53	6	0	93	10	2	Receive	Normal
30/11/2010	08:44:21	2	0	93	10	6	Transmit	Normal
30/11/2010	08:51:55	6	0	93	10	2	Receive	Normal
30/11/2010	08:53:00	2	0	93	10	6	Transmit	Normal
30/11/2010	08:59:10	6	0	93	10	2	Receive	Normal
30/11/2010	08:59:34	2	0	93	10	6	Transmit	Normal
30/11/2010	09:05:44	6	0	93	10	2	Receive	Normal
30/11/2010	09:06:08	2	0	93	10	6	Transmit	Normal
30/11/2010	09:13:14	6	0	93	10	2	Receive	Normal
30/11/2010	09:15:31	2	0	93	10	6	Transmit	Normal
30/11/2010	09:23:59	6	0	93	10	2	Receive	Normal
30/11/2010	09:24:31	2	0	93	10	6	Transmit	Normal
30/11/2010	09:30:13	6	0	93	10	2	Receive	Normal
30/11/2010	09:30:33	2	0	93	10	6	Transmit	Normal
30/11/2010	09:35:21	6	0	93	10	2	Receive	Normal
30/11/2010	09:35:49	2	0	93	10	6	Transmit	Normal
30/11/2010	09:41:23	6	0	93	10	2	Receive	Normal
30/11/2010	09:41:57	2	0	93	10	6	Transmit	Normal
30/11/2010	10:01:18	6	0	93	10	2	Receive	Normal
30/11/2010	10:01:36	2	0	93	10	6	Transmit	Normal
30/11/2010	10:07:50	6	0	93	10	2	Receive	Normal
30/11/2010	10:08:13	2	0	93	10	6	Transmit	Normal
30/11/2010	10:16:01	6	0	93	10	2	Receive	Normal
30/11/2010	10:16:22	2	0	93	10	6	Transmit	Normal
30/11/2010	10:22:12	6	0	93	10	2	Receive	Normal
30/11/2010	10:22:44	2	0	93	10	6	Transmit	Normal
30/11/2010	10:34:37	6	0	93	10	2	Receive	Normal
30/11/2010	10:34:51	2	0	93	10	6	Transmit	Normal
30/11/2010	10:42:17	6	0	93	10	2	Receive	Normal
30/11/2010	10:42:49	2	0	93	10	6	Transmit	Normal
30/11/2010	10:57:39	6	0	93	10	2	Receive	Normal
30/11/2010	10:58:02	2	0	93	10	6	Transmit	Normal
30/11/2010	11:12:13	6	0	93	10	2	Receive	Normal

Links dedicados são divididos por um par de dispositivos que comunicam-se durante um time *slot* alocado. *Links* compartilhados podem ter mais de um remetente mas somente um receptor. *Links* broadcast tem apenas um remetente e muitos que podem escutá-lo. O time *slot* é repetido em uma taxa correspondente ao tamanho do *superframe*. Dependendo do tipo do *link* no time slot o dispositivo pode tomar as seguintes ações:

- Tentativa de transmissão de um pacote;
- Em espera para receber um pacote;
- Ficar em modo ocioso.

Um dispositivo que possui um *link* de transmissão ou um transmissão/recepção deve enviar um pacote para o time *slot* associado se o destino do pacote combinar com o vizinho na outra ponta do *link*. Um dispositivo que tem um *link* de recepção, ou um de transmissão/recepção sem pacotes para enviar, fica na espera de um pacote durante o time *slot* associado.

Na tabela 22 foram ordenados os *links* em função do número do time *slot* e do *superframe* (*superframe* ID 0). Através do no comando 783 é possível verificar que este frame é composto por 1024 *slots*.

Tabela 22: Análise dos *Slots* no *Superframe* 0.

Nickname	*Superframe* ID	*Slot* number(*link*)	Channel *Offset*	Neighbor	*Link* Options	*Link* type
7	0	1	0	2	Transmit|Receive	Discovery
7	0	28	3	2	Transmit	Normal
2	0	29	9	6	Transmit	Normal
2	0	93	10	6	Transmit	Normal
6	0	98	7	1	Transmit	Normal
5	0	146	8	1	Transmit	Normal
7	0	157	8	2	Transmit	Normal
3	0	161	2	1	Transmit	Normal
3	0	180	6	f980	Receive	Join
6	0	221	9	ffff	Receive	Join
4	0	222	5	6	Transmit	Normal

6	0	222	5	4	Receive	Normal
6	0	226	10	1	Transmit	Normal
2	0	349	10	6	Transmit	Normal
5	0	402	5	1	Transmit	Normal
7	0	412	10	2	Transmit	Normal
3	0	417	4	1	Transmit	Normal
7	0	445	11	f980	Receive	Join
4	0	477	3	6	Transmit	Normal
6	0	477	3	4	Receive	Normal
4	0	495	2	ffff	Receive	Join
6	0	577	6	1	Transmit	Normal

Nesta análise é possível observar na próxima figura como os canais são ocupados e é realizada a troca de pacotes. Nos *slots* que foram obtidos seqüencialmente ou próximos (*slot* número 28, 29, 93 e 98) neste fragmento dos dados, por exemplo, fica claro a transferência de pacotes do dispositivo 7 para o 2 do dispositivo 2 para o 6 e finalmente do dispositivo 6 para o gateway (dispositivo 1).

TS / Ch.Offset	1	28	29	93	98	146	157	161	180	221	222	226	349	402	412	417	477
0	7 -> 2	7 -> 2															
1																	
2								3 -> 1									
3																	4 -> 7
4																3 -> 1	
5											4 -> 6			5 -> 1			
6									3 -> f980								
7					6 -> 1												
8						5 -> 1	7 -> 2										
9			2 -> 6							6 -> ffff							
10				2 -> 6								6 -> 1	2 -> 6		7 -> 2		

Figura 49. *Links* entre os Dispositivos.

Os dados obtidos a partir do comando que identifica a lista de links utilizada pelo dispositivo permitiram determinar algumas possíveis rotas.

A partir do comando 800 foram registrados os serviços. Os serviços sempre ocorreram entre os dispositivos e o gateway. Na Tabela 23 um trecho dos dados obtidos, onde *source* é o valor do apelido do dispositivo requisitado. Estes dados não foram utilizados para a análise de rotas.

Tabela 23: Dados da Lista de Serviços dos Dispositivos.

Date	Hour	Source	Active services	Service ID	Service Req. Flag	Service Domain	Nickname Peer	Periodo	Route ID
30/11/2010	10:02:02	4	2	0	1	0	F981	300000.00	1
30/11/2010	10:02:02	4	2	128	3	2	F981	60000.00	2
30/11/2010	10:02:16	6	2	0	1	0	F981	60000.00	1
30/11/2010	10:02:16	6	2	128	3	2	F981	60000.00	2
30/11/2010	10:02:20	3	2	128	3	2	F981	60000.00	2
30/11/2010	10:02:20	3	2	1	1	0	F981	1000.00	1
30/11/2010	10:02:33	5	2	128	3	2	F981	60000.00	2
30/11/2010	10:02:33	5	2	1	1	0	F981	1000.00	1
30/11/2010	10:02:37	2	2	128	3	2	F981	60000.00	2
30/11/2010	10:02:37	2	2	1	1	0	F981	1000.00	1
30/11/2010	10:02:47	7	1	128	3	2	F981	60000.00	2
30/11/2010	10:08:45	4	2	0	1	0	F981	300000.00	1

As sessões gerenciam na camada de rede as comunicações entre dois dispositivos ou entre grupo de dispositivos. Todos dispositivos possuem duas sessões com o gerenciador de rede (F980): uma para comunicação entre pares, e outra para uso de difusão da rede (*broadcast*) do gerenciador de rede. Ainda também possuem duas sessões com o gateway (F980). A saída de resposta dos dados sobre as sessões dos dispositivos mostrou-se de acordo com a especificação HART e apresenta-se no formato abaixo:

Tabela 24: Formato dos dados registrado sobre as Sessões dos dispositivos.

Date	Hour	Nickname	Number Active Sessions	Session Type	Peer Device Nick	Peer Device Unique ID	Peer Device Nonce CV	Device Nonce CV
30/11/2010	08:35:09	4	4	Unicast	F980	F980000001	13	11
30/11/2010	08:35:09	4	4	Broadcast	F980	F980000001	24	11
30/11/2010	08:35:09	4	4	Broadcast	F981	F981000002	1	17
30/11/2010	08:35:09	4	4	Unicast	F981	F981000002	22	17
30/11/2010	08:35:29	6	4	Broadcast	F980	F980000001	22	147
30/11/2010	08:35:29	6	4	Unicast	F980	F980000001	28	147
30/11/2010	08:35:29	6	4	Broadcast	F981	F981000002	1	1251
30/11/2010	08:35:29	6	4	Unicast	F981	F981000002	443	1251
...	...	...	...	...	...	...	...	...
30/11/2010	11:18:02	2	4	Unicast	F981	F981000002	309	445
30/11/2010	11:18:42	7	4	Broadcast	F981	F981000002	1	1
30/11/2010	11:18:42	7	4	Broadcast	F980	F980000001	27	1

30/11/2010	11:18:42	7	4	Unicast	F980	F980000001	29	33
30/11/2010	11:18:42	7	4	Unicast	F981	F981000002	128	103

Outros dados obtidos são referentes às identificações de rotas através dos grafos ou da fonte. A tabela 21 apresenta os dados obtidos em uma resposta do gateway, neste intervalo todos dispositivos disponíveis estavam associados à rede e os dados registrados mantiveram-se constantes durante todo o período de análise, com exceção quando um dispositivo desconectou-se da rede. É possível verificar valores nulos tanto para a identificação do grafo como para a rota pela origem, esses valores não foram os esperados uma vez que o protocolo WirelessHART define para multi saltos um dos dois métodos para roteamento. Outra particularidade válida nestes dados é quanto ao número de rotas ativas que foram registradas neste ensaio. Os dispositivos comerciais utilizados neste ensaio sempre apresentaram número de rotas igual a 3 enquanto os outros dispositivos mantiveram-se em 4 rotas ativas, diferenciando em uma rota a mais para o gerenciador de rede.

Tabela 25: Identificação de Rotas e Grafos.

Date	Hour	Number Active Routes	Source	Destination	Route ID	Graph ID	Source Route(0)
30/11/2010	09:58:08	3	4	F980	0	0	0
30/11/2010	09:58:08	3	4	F981	1	0	0
30/11/2010	09:58:08	3	4	F981	2	0	0
30/11/2010	09:58:16	3	6	F980	0	0	0
30/11/2010	09:58:16	3	6	F981	1	0	0
30/11/2010	09:58:16	3	6	F981	2	0	0
30/11/2010	09:58:22	4	3	F980	254	0	0
30/11/2010	09:58:22	4	3	F980	0	0	0
30/11/2010	09:58:22	4	3	F981	1	0	0
30/11/2010	09:58:22	4	3	F981	2	0	0
30/11/2010	09:58:28	4	5	F980	254	0	0
30/11/2010	09:58:28	4	5	F980	0	0	0
30/11/2010	09:58:28	4	5	F981	1	0	0
30/11/2010	09:58:28	4	5	F981	2	0	0
30/11/2010	09:58:42	4	2	F980	254	0	0
30/11/2010	09:58:42	4	2	F980	0	0	0
30/11/2010	09:58:42	4	2	F981	1	0	0
30/11/2010	09:58:42	4	2	F981	2	0	0

30/11/2010	09:59:00	4	7	F980	254	0	0
30/11/2010	09:59:00	4	7	F980	0	0	0
30/11/2010	09:59:00	4	7	F981	1	0	0
30/11/2010	09:59:00	4	7	F981	2	0	0

A partir do comando 840 do protocolo WirelessHART é possível obter diversos dados sobre o comportamento dinâmico da rede. Dados dessa ordem são de grande importância para permitir que o gerenciador de rede possa determinar e formar a topologia de rede mais adequada entre os diversos dispositivos.

Para este comando as respostas também não foram satisfatórias, apesar de alguns campos de resposta apresentarem-se coerentes com o comportamento dos dispositivos na rede outros apresentaram valores inadequados e inconsistentes. O Comando 840 é um comando direto com o gateway e retorna 56 bytes de dados estatísticos para cada dispositivo requisitado. Uma possibilidade da falha dos dados recebidos pode ser um erro na implementação deste comando no firmware do Gateway Emerson 1420A.

Date	Hour	Unique ID	Nickname	Active Graph	Active Frames	Active Links	Neighbors	Joins	Date recent join	Time recently joined	Packets generated	Packets terminated	Data-Link layer failures	Network layer failure	CRC errors	Nonce Counter by	Nonce Counter from	Standard Deviation
30/11/2010	08:30:48	26587a29e0	4	0	0	0	1	2	30/11/2010	1198636192	0	0	0	0	0	0	0	0
30/11/2010	08:31:00	26587a2c69	6	0	0	0	3	1	29/11/2010	2407245696	0	0	0	0	0	0	0	0
30/11/2010	08:31:04	e0ff000503	3	0	0	0	3	5	30/11/2010	1194806592	0	0	0	0	0	0	0	0
30/11/2010	08:31:08	e0ff000602	5	0	0	0	2	2	30/11/2010	1181371840	0	0	0	0	0	0	0	0
30/11/2010	08:36:26	26587a29e0	4	0	0	0	1	2	30/11/2010	1198636192	0	0	0	0	0	0	0	0
30/11/2010	08:36:30	26587a2c69	6	0	0	0	3	1	29/11/2010	2407245696	0	0	0	0	0	0	0	0
30/11/2010	08:36:34	e0ff000503	3	0	0	0	3	5	30/11/2010	1194806592	0	0	0	0	0	0	0	0
30/11/2010	08:36:40	e0ff000602	5	0	0	0	2	2	30/11/2010	1181371840	0	0	0	0	0	0	0	0
30/11/2010	08:40:49	26587a29e0	4	0	0	0	1	2	30/11/2010	1198636192	0	0	0	0	0	0	0	0
30/11/2010	08:41:03	26587a2c69	6	0	0	0	4	1	29/11/2010	2407245696	0	0	0	0	0	0	0	0
30/11/2010	08:41:07	e0ff000503	3	0	0	0	3	5	30/11/2010	1194806592	0	0	0	0	0	0	0	0
30/11/2010	08:41:14	e0ff000602	5	0	0	0	2	2	30/11/2010	1181371840	0	0	0	0	0	0	0	0
30/11/2010	08:49:22	26587a2c69	6	0	0	0	4	1	29/11/2010	2407245696	0	0	0	0	0	0	0	0
30/11/2010	08:49:26	e0ff000503	3	0	0	0	4	5	30/11/2010	1194806592	0	0	0	0	0	0	0	0
30/11/2010	08:49:32	e0ff000602	5	0	0	0	2	2	30/11/2010	1181371840	0	0	0	0	0	0	0	0
30/11/2010	08:49:50	e0ff000404	2	0	0	0	2	1	30/11/2010	1232817024	0	0	0	0	0	0	0	0
30/11/2010	08:56:36	26587a29e0	4	0	0	0	3	2	30/11/2010	1198636192	0	0	0	0	0	0	0	0
30/11/2010	08:56:40	26587a2c69	6	0	0	0	4	1	29/11/2010	2407245696	0	0	0	0	0	0	0	0
30/11/2010	08:56:46	e0ff000503	3	0	0	0	4	5	30/11/2010	1194806592	0	0	0	0	0	0	0	0
30/11/2010	08:56:50	e0ff000602	5	0	0	0	2	2	30/11/2010	1181371840	0	0	0	0	0	0	0	0
30/11/2010	08:57:00	e0ff000404	2	0	0	0	2	1	30/11/2010	1232817024	0	0	0	0	0	0	0	0

Figura 50. Fragmento Dados obtidos Comando 840 WirelessHART.

8.4.2 TOPOLOGIA OBTIDA

Através do ensaio realizado e interpretação dos dados obtidos verificou-se uma possível topologia para a distribuição dos dispositivos. Essa topologia foi registrada avaliando-se os dados de quando todos os dispositivos estavam associados à rede sendo que

os principais parâmetros utilizados foram os vizinhos descobertos e os links estabelecidos entre os dispositivos, estas variáveis estão associados aos comandos 780 e 784 respectivamente.

Os dispositivos 2, 4 e 7 utilizaram dispositivos intermediários para estabelecer e trocar pacotes de dados com o gateway. Os caminhos utilizados são demonstrados na Figura 51.

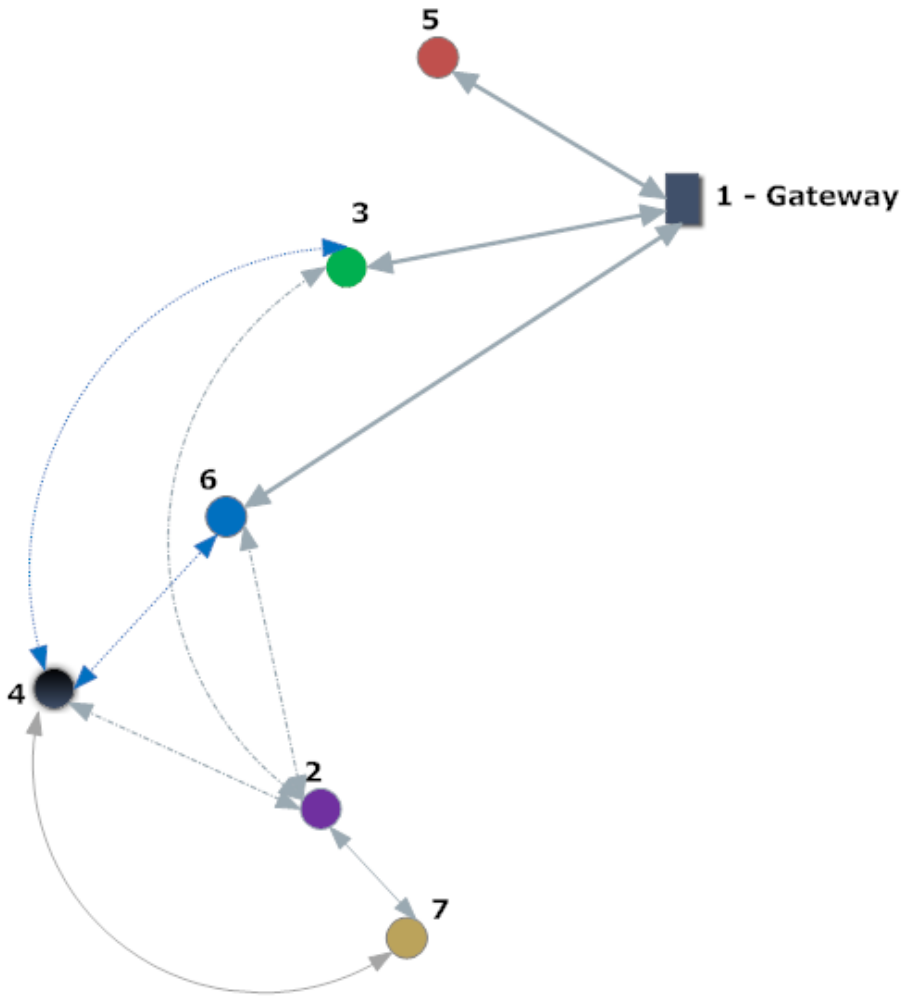

Figura 51. Topologia construída.

Dentro da análise de rotas pela interpretação dos links estabelecidos ainda foi identificado características pertinentes do protocolo como o número máximo de bytes no frame físico o qual limitou a resposta da requisição do comando 784 em 10 links. No entanto maior quantidade de links registrados deve permitir também uma análise mais precisa sobre o

roteamento dos pacotes de dados entre os dispositivos da rede. Uma possível solução é a modificação da rotina do software implementado para o comando 784 que possibilite realizar uma leitura de todos os links na lista de cada dispositivo.

9 RESULTADOS ALCANÇADOS

Este projeto propôs o desenvolvimento de um software o qual permitisse diagnosticar as características da topologia formada entre os dispositivos WirelessHART. No projeto foi possível implementar rotinas computacionais que através de comunicação com o gateway permitiram a aquisição de dados referente a algumas variáveis envolvidas na natureza do protocolo WirelessHART. Conseguiu-se 4 horas de registros referentes a uma rede onde apenas alguns dispositivos possuíam comunicação com o gateway. A obtenção destes dados foi válida e permitiu determinar possíveis rotas de comunicação entre os dispositivos utilizados, ainda é possível avaliar que determinados dispositivos perderam conexão com a rede. É viável na continuidade deste trabalho considerar de forma mais precisa os detalhes de comunicação registrados que circundam estes eventos permitindo assim eliminar possibilidades errôneas e seguir para a identificação da causa de falha de uma forma mais precisa.

Foram realizadas diversas tentativas de ensaios, sendo que houve grande dificuldade de obter uma rede de dispositivos que estabelecessem comunicação utilizando nodos intermediários em geral ou os dispositivos possuíam comunicação direta com o gateway ou não se conectavam. Outra dificuldade foi a pouca quantidade de rádios disponíveis e confiáveis para a realização dos testes. Um teste mais interessante deve disponibilizar mais dispositivos para que possam surgir diferentes rotas de comunicação (diferentes tipos de roteamento) para assim buscar resultados mais completos quanto a dinâmica da rede WirelessHART.

Os resultados obtidos foram pertinentes e permitem estudos futuros principalmente quanto a validade dos equipamentos utilizados no laboratório assim como no desenvolvimento da pilha WirelessHART para estes equipamentos.

10 CONCLUSÃO

O controle de processos industriais tem passado por diversos avanços tecnológicos. De todas as tecnologias associadas à indústria, as redes de comunicação são as que passaram por maiores mudanças. Com um histórico que parte desde a comunicação pneumática até a então era das comunicações sem fio, o controle de processos na indústria vem aumentando o grau de automatização possibilitando maior quantidade de dados, maior atuação e eficiência no controle de processos, mas também em muitos casos aumentou também a necessidade de maior utilização de cabos, como por exemplo, nas ligações paralelas.

A tecnologia de comunicação sem fio voltada para a indústria já apresentou algumas tentativas de aceitação no mercado, no entanto, além da necessidade de qualidade no protocolo utilizado como robustez, flexibilidade e confiabilidade, ainda existem interesses comerciais os quais podem impedir os avanços de determinada tecnologia. O padrão WirelessHART possui grandes corporações além da própria Fundação HART que impulsionam o desenvolvimento e aceitação desta tecnologia na indústria. O protocolo de comunicação WirelessHART foi certificado como o primeiro padrão de comunicação sem fio para controle de processos pela IEC (*International Electrotechnical Commission),* baseado em redes de malha esse protocolo demonstra características que trazem maior confiabilidade e flexibilidade no controle de processos. O WirelessHART ainda é um protocolo novo apesar de sua construção ser baseada no protocolo HART, o qual é um protocolo de comunicação consagrado na indústria.

Com a certificação do protocolo WirelessHART diversos fabricantes de equipamentos para automação e processos de controle estão buscando recursos para obter espaço no mercado de dispositivos com tecnologia sem fio. Essa corrida pelo mercado pode trazer algumas conseqüências negativas, uma delas é a comercialização de equipamentos os quais ainda não conseguem garantir todas as exigências da especificação do protocolo.

Este trabalho apresenta o desenvolvimento de uma ferramenta que possibilita obter diversos dados dos equipamentos em operação que compõem uma rede WirelessHART. Inicialmente buscou-se o monitoramento da dinâmica da topologia da rede, logo que uma das características de destaque deste padrão sem fio é a capacidade de formar topologia em malha e caminhos redundantes não sendo necessário mais gateways ou roteadores apesar destes últimos diminuírem a latência de comunicação entre os equipamentos. Na seqüência dos ensaios observou-se algumas possíveis divergências para alguns dados auferidos o que direcionou também esta pesquisa para a avaliação da integridade das informações recebidas por parte dos equipamentos utilizados.

Os ensaios realizados permitiram observar algumas características do protocolo proporcionando maior entendimento do comportamento da rede WirelessHART, ainda foi possível verificar que os equipamentos utilizados nos ensaios apresentaram comportamentos diferentes para algumas variáveis registradas, essas questões são relevantes e devem auxiliar na continuidade de pesquisas com WirelessHART assim como no desenvolvimento da pilha WirelessHART utilizada nos equipamentos de desenvolvimento presentes nos ensaios.

Através destes ensaios e resultados é possível partir para um próximo passo no desenvolvimento desta ferramenta. Como perspectiva de trabalhos futuros é válido aperfeiçoar esta aplicação e como medidas iniciais pode-se excluir da requisição alguns dados que não foram tão relevantes em função do tipo de análise desejada, permitindo assim obtenção de maior desempenho no software. Ainda é válido e fica para trabalhos futuros o desenvolvimento de uma interface gráfica que permita uma interação mais amigável com o usuário.

11 REFERÊNCIAS

[1] Hart (Ed.) **HART Communication Foundation**, **A Technical Overview**, HCF_LIT-20 Revision 3.0, 2007.

[2] HART Communication. Disponível online em: http://www.hartcomm2.org/index.html.

[3] SPRINGER (Ed). **WirelessHART** : real-time mesh network for industrial automation. (S.1.): 2010.

[4] IEEE 802.11. Disponível online em: http://grouper.ieee. org/groups/802/11/.

[5] Hart (Ed.) **HART Communication Foundation**, Network Management Specification, HCF_SPEC-085 Revision 1.1, 2008.

[6] Theodore S. Rappaport, "Wireless Communications – Principles & Practice," Prentice Hall Communications Engineering and Emerging Technologies Series, 1996.

[7] Hart (Ed.) **HART Communication Foundation**, TDMA Data Link Layer Specification, HCF_SPEC-075 Revision 1.1, 2008.

[8] Hart (Ed.) **HART Communication Foundation**, Addendum to Wireless Command Specification, HCF_SPEC-155 Revision 7.0, 2008.

[9] Hart (Ed.) **HART Communication Foundation**, Command Summary Specification, HCF_SPEC-099 Revision 9.0, 2007.

[10] Hart (Ed.) **HART Communication Foundation**, Wireless Command Specification, HCF_SPEC-155 Revision 1.0, 2007.

[11] Hart (Ed**.) HART Communication Foundation,** Wireless Introduction, HCF_LIT-131 Revision 1.0 [S.1.], 2010.

[12] Hart (Ed.) **HART Communication Foundation**, Wireless Devices Specification, HCF_SPEC-290 Revision 1.0, 2007.

[13] Hart (Ed.) **HART Communication Foundation**, Command Response Code Specification, HCF_SPEC-307 Revision 6.0, 2007.

[14] SHARP, J. **Visual C# 2008,** Step by Step. Microsoft (Ed.), 2008.

[15] Hart (Ed.) **HART Communication Foundation**, WirelessHART Device Types - Gateways, HCF_LIT-119 Revision 1.0, 2010.

[16] Rosemont Reference Manual (Ed.) **648 Wireless Temperature Transmitter**. Revision BA, 2007.

[17] MULLER, Ivan; PEREIRA, Carlos E.; NETTO, João C.; FABRIS, Eric C.; ALGAYER, Rodrigo; **Development of WirelessHART Compatible Field Devices**. 2010.

[18] FREESCALE, MC1322x. Technical Data. Revision 1.3, 2010.

ANEXO A – DETALHES DOS COMANDOS UTILIZADOS PARA AQUISIÇAO DE DADOS

Comando 780

Byte	Formato	Descrição
0	Unsigned-8	Índice tabela de vizinhos
1	Unsigned-8	Entrada de número de vizinhos

Byte	Formato	Descrição
0	Unsigned-8	Índice tabela de vizinhos
1	Unsigned-8	Entrada de número de vizinhos
2	Unsigned-8	Número total de vizinhos
3-4	Unsigned-16	Apelido do vizinho (2 bytes Apelido)
5	Bit - 8	Grafo ID
6	Signed-8	Gráfico ID n (baseado no número de ID recebido)
7-8	Unsigned-16	Número de pacotes transmitidos para este vizinho
9-10	Unsigned-16	Número de falhas na transmissão
11-12	Unsigned-16	Número de pacotes recebidos deste vizinho
13-…		Número de entradas baseado na resposta do byte 1

Código	Classe	Descrição
0	Sucesso	Sem erros de comando específico
1		Indefinido
2	Erro	Seleção inválida
3-4		Indefinido
5	Erro	Poucos dados recebidos
6-7		Indefinido
8	Aviso	Ajustado para o valor mais próximo
9-127		Indefinido

Comando 782

Byte	Formato	Descrição
0	Unsigned-8	Índice de sessões
1	Unsigned-8	Número de entradas para leitura

Byte	Formato	Descrição
0	Unsigned-8	Índice de sessões
1	Unsigned-8	Número de entradas para ler
2	Unsigned-8	Número de sessões ativas
3	Enum-8	Tipo de sessões
4-5	Unsigned-16	Nickname do dispositivo par
6-10	Unsigned-40	Identificação Única do par
11-14	Unsigned-32	Valor único do par de dispositivos (nonce)
15-18	Unsigned-32	Valor único do dispositivo (nonce)
19-34		3-18 repete para o número de entradas indicado na resposta do byte 1

Código	Classe	Descrição
0	Sucesso	Sem erros de comando específico
1-4		Indefinido
5	Erro	Seleção inválida
6-7		Indefinido
8	Aviso	Ajustado para o valor mais próximo
9-127		Indefinido

Comando 783

Byte	Formato	Descrição
0	Unsigned-8	Índice de *superframe*s
1	Unsigned-8	Número de entradas para leitura

Byte	Formato	Descrição
0	Unsigned-16	Índice de *superframe*s
1	Unsigned-8	Número de entradas para leitura
2	Unsigned-16	Número de *superframe*s ativos
3	Unsigned-8	*Superframe* ID
4-5	Unsigned-16	Número de *slots* neste *superframe*
6	Signed-8	Flags modo *superframe*s
7-10	Unsigned-16	Resposta dos bytes 3 - 6 repetidos baseado na resposta do byte 1

Código	Classe	Descrição
0	Sucesso	Sem erros de comando específico
1		Indefinido
2	Erro	Seleção inválida
3-4		Indefinido
5	Erro	Poucos dados recebidos
6-7		Indefinido
8	Aviso	Ajustado para o valor mais próximo
9-127		Indefinido

Comando 784

Byte	Formato	Descrição
0	Unsigned-8	Índice de *links*
1	Unsigned-8	Número de *links* para leitura

Byte	Formato	Descrição
0-1	Unsigned-16	Índice de *links*
2	Unsigned-8	Número de *links* para leitura
3-4	Unsigned-16	Número de *links* ativos
5	Unsigned-8	*Superframe* ID
6-7	Unsigned-16	Número do *slot* no *superframe* para este *link*
8	Signed-8	Canal de compensação para este *link*
9-10	Unsigned-16	Apelido do vizinho para este *link* (ou 0xFFFF *broadcast link*)
11	Bits-8	Opções de *links*
12	Enum-8	Tipos de *links*
13-…		Resposta dos bytes 5-12 dependendo da resposta do byte 2

Código	Classe	Descrição
0	Sucesso	Sem erros de comando específico
1		Indefinido
2	Erro	Seleção inválida
3-4		Indefinido
5	Erro	Poucos dados recebidos
6-7		Indefinido
8	Aviso	Ajustado para o valor mais próximo
9-127		Indefinido

Comando 800

Byte	Formato	Descrição
0	Unsigned-8	Índice de serviços
1	Unsigned-8	Número de entradas para leitura

Byte	Formato	Descrição
0	Unsigned-8	Índice de serviço
1	Unsigned-8	Número de entradas para ler
2	Unsigned-8	Número de serviços ativos
3	Unsigned-8	Identificação de serviço (ID)
4	Signed-8	Flag de requisição de serviço
5	Enum-8	Dominio da aplicação de serviço(0-publish;1-event;2-maintenance;3-block transfer
6-7	Unsigned-16	Apelido do par com o qual o serviço é requisitado
8-11	Time	Período (latência se flag intermitente é ajustado)
12	Unsigned-8	Route ID
13...		Repete conforme entrada no byte 1

Código	Classe	Descrição
0	Sucesso	Sem erros de comando específico
1		Indefinido
2	Erro	Seleção inválida
3-4		Indefinido
5	Erro	Poucos dados recebidos
6-7		Indefinido
8	Aviso	Ajustado para o valor mais próximo
9-127		Indefinido

Comando 802

Byte	Formato	Descrição
0	Unsigned-8	Índice de rotas
1	Unsigned-8	Número de entradas para leitura

Byte	Formato	Descrição
0	Unsigned-8	Índice de rotas
1	Unsigned-8	Número de entradas para ler
2	Unsigned-8	Número de rotas ativas
3	Unsigned-8	Número de rotas remanescentes
4	Unsigned-8	Identificação de rotas
5-6	Unsigned-16	Apelido de destino
7-8	Unsigned-16	Identificação do grafo
9	Unsigned-8	Source route (1=attached,0=none)
10...		4-9 repete para o número de entradas indicado na resposta do byte 1

Código	Classe	Descrição
0	Sucesso	Sem erros de comando específico
1		Indefinido
2	Erro	Seleção inválida
3-4		Indefinido
5	Erro	Poucos dados recebidos
6-7		Indefinido
8	Aviso	Ajustado para o valor mais próximo
9-127		Indefinido

Comando 840

Byte	Formato	Descrição
0-4	Unsigned-40	Identificação do dispositivo

Byte	Formato	Descrição
0-4	Unsigned-40	Identificação do dispositivo
5-6	Unsigned-16	Número de grafos ativos
7-8	Unsigned-16	Número de frames ativos
9-10	Unsigned-16	Número de *links* ativos
11	Bit - 8	Número de vizinhos
12-15	Time	Latência média de comunicação do gateway para este nó
16-17	Unsigned-16	Número de associações (join)
18-20	Date	Data da última associação
21-24	Unsigned-32	Tempo em 1/32 de ms
25-28	Unsigned-32	Número de pacotes gerados por este dispositivo
29-32	Unsigned-32	Número de pacotes terminados por este dispositivo
33-36	Unsigned-32	Número de falhas detectadas na camada de enlace
37-40	Unsigned-32	Número de falhas detectadas na camada de rede (sessões)
41-44	Unsigned-32	Números de erros checados no CRC
45-48	Unsigned-32	Número do *Nonce Counter* não recebido por este dispositivo
49-52	Unsigned-32	Número do *Nonce Counter* não recebido do dispositivo
53-56	Time	Desvio padrão da latência

Código	Classe	Descrição
0	Sucesso	Sem erros de comando específico
1		Indefinido
2	Erro	Seleção inválida
3-4		Indefinido
5	Erro	Poucos dados recebidos
6-127		Indefinido