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

ALEXANDRE BENTO LEAL

Ferramenta para análise de logs de equipamentos de teste de conformidade, usados com o protocolo WirelessHART

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

Orientador: Prof. Dr. Sérgio L. Cechin
Co-orientador: Prof. Dr. João Cesar Netto

Porto Alegre
2016

AGRADECIMENTOS

Agradeço a todos que de alguma forma contribuíram para a conclusão deste trabalho, em especial aos meus pais Arnaldo e Martha e aos meus irmãos Luís Henrique e Márcia.

Aos amigos e colegas do curso de Engenharia de Computação Felipe Ávila Nesello, Renato Westphal e André Capitani Gusmão.

Aos colegas e amigos do Laboratório de Automação, Sistemas de Controle e Robótica, em especial ao Jean Michel Winter e ao Matheus Maia de Souza que auxiliaram com muita prestatividade.

Ao Professor João Cesar Netto, e ao professor Sérgio L. Cechin que contribuiu fortemente para o desenvolvimento deste trabalho.

RESUMO

Os testes de verificação de conformidade de dispositivos à norma WirelessHART são fundamentais para o correto funcionamento dos mesmos. Porém, os equipamentos de teste apresentam arquivos de *log* independentes, com informações em demasia, tornando trabalhosa a análise dos testes. A possibilidade de criar um único arquivo de *log*, unindo apenas informações pertinentes à análise dos testes e com sincronização temporal aceleraria e facilitaria o processo. Este trabalho propõe e implementa uma ferramenta que seleciona, une, organiza e pré-analisa algumas informações do teste, gerando um arquivo de *log* resultante único. O trabalho foi desenvolvido juntamente com a equipe do LASCAR (Laboratório de Automação, Sistemas de Controle e Robótica), da UFRGS. Ao final, os resultados mostraram que a ferramenta funcionou adequadamente para o objetivo proposto e tornou-se útil para a análise dos testes.

Palavras-chave: WirelessHART, Testes de Verificação de Conformidade, Wi-Analys, Wi-HTest.

ABSTRACT

WirelessHART devices compliance verification tests are essential for the correct functioning. Os testes de verificação de conformidade de dispositivos à norma WirelessHART são fundamentais para o correto funcionamento dos mesmos. However, test equipment has independent log files with too much information, making tests' analysis laborious. The possibility of creating a single log file, joining only relevant information to tests' analysis and with time synchronization would speed up and facilitate the process. This work proposes and implements a tool that selects, combines, organizes and pre-analyzes some test information, generating a single resulting log file. The work was developed with the LASCAR team (Laboratório de Automação, Sistemas de Controle e Robótica), from UFRGS. Finally, results showed that the tool worked adequately for the proposed aim and became useful for the analysis of the tests.

Keywords: WirelessHART, Compliance Tests, Wi-Analys, Wi-HTest.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AES	*Advanced Encryption Standard*
ASN	*Absolute Slot Number*
CBC-MAC	*Cipher Block Chaining Message Authentication Code*
CCM	*Counter with CBC-MAC*
CMD	*Command*
CRC	*Cyclic Redundancy Check*
DC	Dispositivo de Campo
DET	Dispositivo em Teste
DLL	*Data Link Layer*
FSK	*Frequency Shift Keying*
HCF	*HART Communication Foundation*
MAC	*Medium Access Control*
MIC	*Message Integrity Code*
PDU	*Protocol Data Unit*
PPS	*Post Processing Suite*
PWH	Protocolo WirelessHART
TDMA	*Time Division Multiple Access*
TVC	Teste de Verificação de Conformidade

SUMÁRIO

1 INTRODUÇÃO

Gerações de avanços tecnológicos impactaram a comunicação industrial. O controle de processos industriais evoluiu do controle centralizado para o controle distribuído. Estações de trabalho do operador são o centro dos sistemas de controle distribuído atuais. Elas conectam as operações de planta ao processo. Acompanhando a chegada de cada nova geração de produtos, as estações têm se tornado cada dia mais inteligentes. Aplicações inovadoras fornecem alarme, controle e diagnóstico. Por trás dessas aplicações estão os dispositivos inteligentes. Uma grande variedade de dispositivos inteligentes surgiu com avanços no desenvolvimento de sensores e diagnósticos. Os dispositivos costumam oferecer diagnóstico de linhas cruzadas, vazamentos, desgaste de orifício, entre outros. Surgiu o desafio de conectar as funcionalidades desses dispositivos inteligentes com suas infraestruturas de sistema de controle.

A indústria esforça-se em especificar protocolos de comunicação desde a década de 80. Eles devem atender a requisitos de processo como confiabilidade, segurança e durabilidade. Redes cabeadas trazem desvantagens como o difícil acesso confiável a todas partes da planta, alto custo de material e de trabalho, além da difícil instalação. Tentou-se o uso industrial de protocolos de comunicação *wireless* como WINA, Zigbee, WiFi e *Bluetooth*, que trouxeram problemas de alto consumo ou robustez e não foram adotados em larga escala. Condições adversas como temperaturas extremas e altas radiações são comuns em ambientes industriais. Portanto, segurança e disponibilidade devem ser oferecidos por dispositivos e protocolos de comunicação que compõem os sistemas industriais.

O WirelessHART é um protocolo de comunicação para ambientes industriais que foi desenvolvido com as características que o ambiente exige. Ele oferece segurança, robustez, simplicidade, baixo consumo e compatibilidade com o padrão HART. Dispositivos fabricados para funcionar com o padrão HART embarcado devem passar por testes de conformidade. Atualmente, os testes são executados e interpretados de maneira manual pelo analista, o que torna o processo lento. Os relatórios gerados pelo fabricante são confusos e com informações em demasia, dificultando o diagnóstico dos testes.

Este trabalho apresenta uma ferramenta que facilita e agiliza a verificação de conformidade dos dispositivos à norma WirelessHART. Esta ferramenta objetiva filtrar, reunir e interpretar informações geradas pelos relatórios do fabricante de modo a facilitar o diagnóstico do teste por parte do analista.

2 WIRELESSHART

O protocolo de comunicação WirelessHART originou-se da evolução do protocolo HART.

2.1 Protocolo HART

O protocolo HART (*Highway Addressable Remote Transducer*), em português, Via de Dados Endereçável por Transdutor Remoto, é uma implementação mais recente do *Fieldbus*, um protocolo de automação industrial digital. A sua maior vantagem é que pode comunicar através da fiação analógica legada de 4-20mA (sinal variando de 4 a 20mA), compartilhando os pares de fios utilizados pelo sistema anterior. De acordo com AUTOMATION (2005), devido a larga adoção dos sistemas 4-20mA por todo o mundo, o Protocolo HART é um dos protocolos industriais mais populares atualmente. A tecnologia HART é de fácil utilização e extremamente confiável quando empregada no comissionamento e na calibração de dispositivos inteligentes, bem como em diagnósticos on*line* contínuos.

Existem diversas razões para se colocar um dispositivo hospedeiro em comunicação com dispositivos inteligentes. Dentre elas, destacam-se:

- Configuração ou reconfiguração de dispositivos
- Diagnósticos de dispositivos
- Identificação e resolução de problemas com dispositivos
- Leitura dos valores de medição adicionais fornecidos pelo dispositivo
- Saúde e estado de dispositivos

Os anos de sucesso no aproveitamento dessas vantagens explicam por que a tecnologia HART é o maior de todos os protocolos de comunicação, instalado em mais de 30 milhões de dispositivos em todo o mundo (WIKIPEDIA, 2016).

Empresas do ramo químico-petrolífero como Shell, Monsanto e Statoil, utilizam o HART largamente.

O protocolo HART existe desde meados dos anos 80. Na sua versão inicial, o protocolo foi sobreposto a um sinal 4-20mA fornecendo comunicação em dois sentidos

com dispositivos de campo inteligentes sem comprometer a integridade dos dados medidos. Ao longo de seus 20 anos de existência, o protocolo HART evoluiu de um simples protocolo baseado no sinal 4-20mA para a tecnologia atual baseada na comunicação por cabo e *wireless* com extensas características dando suporte a segurança, transferências de dados não solicitadas, notificações de eventos, transferências de modo de bloco (traduza block mode transfers) e diagnósticos avançados. Os diagnósticos agora incluem informação sobre o dispositivo, o equipamente ao qual o dispositivo está acoplado, e em alguns casos o processo real sendo monitorado. A versão mais recente do protocolo HART, a versão 7, introduz diversas características novas fornecendo performance e diagnóstico melhorados, e capacidades de manutenção melhores.

O protocolo HART versão 7:

- inclui redes *wireless mesh*,
- adiciona sincronização temporal e e *time stamping* dos dados,
- melhora as mensagens (em modo rajada) em *publish/subscribe*s,
- adiciona a camada de transporte,
- adiciona a camada de rede,
- adiciona *pipes* para transferência de arquivo em alta velocidade, e
- adiciona segurança/encriptação/decriptação

(CHEN; NIXON; MOK, 2010, p. 22).

2.2 Visão Geral

O protocolo HART é o padrão global para envio e recebimento de informações digitais através de cabos analógicos, entre dispositivos inteligentes e sistemas de controle ou monitoramento. Mais especificamente, o HART é um protocolo de comunicação bidirecional que possibilita o acesso a dados entre dispositivos de campo inteligentes e sistemas hospedeiros (centralizado). O hospedeiro pode ser qualquer aplicativo de *software*, desde um dispositivo portátil ou *laptop* utilizado pelo técnico até um sistema de segurança, gerenciamento de ativos ou controle de processos da fábrica, ou outro sistema que utilize qualquer plataforma de controle (HARTCOMM, 2014a).

O Protocolo HART faz uso do padrão Bell 202 *Frequency Shift Keying* (FSK) para sobrepor sinais digitais a um nível baixo dentro do intervalo de 3-20mA. Isso permite que aconteça a comunicação em dois sentidos e torna possível de se obter

informações adicionais além da variável de processo normal ser comunicada para/de um instrumento de campo inteligente.

O Protocolo HART transmite dados a uma taxa de 1200bps sem interromper o sinal de 4-20 mA e permite que um aplicativo hospedeiro (mestre) obtenha duas ou mais atualizações digitais por segundo a partir de um dispositivo de campo inteligente. Como o sinal digital FSK é de fase contínua, não há interferência com o sinal de 4-20 mA.

A tecnologia HART é um protocolo mestre/escravo, o que significa que um dispositivo de campo inteligente (escravo) só fala quando chamado por um mestre. O protocolo HART pode ser utilizado em vários modos, tais como ponto-a-ponto ou multiponto para comunicar de/para dispositivos de campo inteligentes e de controle central ou sistemas de monitoramento.

A comunicação HART ocorre entre dois dispositivos HART-habilitados, tipicamente um dispositivo de campo inteligente e um sistema de controle ou monitoramento. A comunicação ocorre usando o fio de grau de instrumentação padrão e utilizando práticas de terminação e fios padronizados.

O protocolo HART fornece dois canais de comunicação simultâneos: o sinal analógico de 4-20 mA e um sinal digital. O sinal de 4-20 mA comunica o valor principal medido (no caso de um dispositivo de campo) usando o *loop* de corrente de 4-20 mA - o mais rápido e confiável padrão da indústria. Informações de dispositivos adicionais são comunicadas utilizando um sinal digital que está sobreposto no sinal analógico.

O sinal digital contém informações do dispositivo, incluindo o estado do dispositivo, diagnósticos, valores adicionais medidos ou calculados, etc. Juntos, os dois canais de comunicação fornecem uma solução completa de comunicação de baixo custo e muito robusta, que é fácil de usar e configurar (HARTCOMM, 2014b).

À medida que a necessidade por novas medições de processos aumentava, os usuários buscavam um método simples, confiável, seguro, eficaz e econômico para transmitir valores de medição aos sistemas de controle sem a necessidade de instalar mais fios. Em virtude de melhorias nos processos, ampliações de instalações, requisitos normativos e exigências de níveis de segurança para medições adicionais, os usuários passaram a considerar a tecnologia *wireless* para essas soluções (HARTCOMM, 2014c).

Assim surgiu o WirelessHART, compatível com o padrão HART e outras tecnologias*wireless*, tendo reduzidos custos de instalação e baixo consumo de energia. O protocolo provê o monitoramento e controle de processos e gerenciamento de ativos.

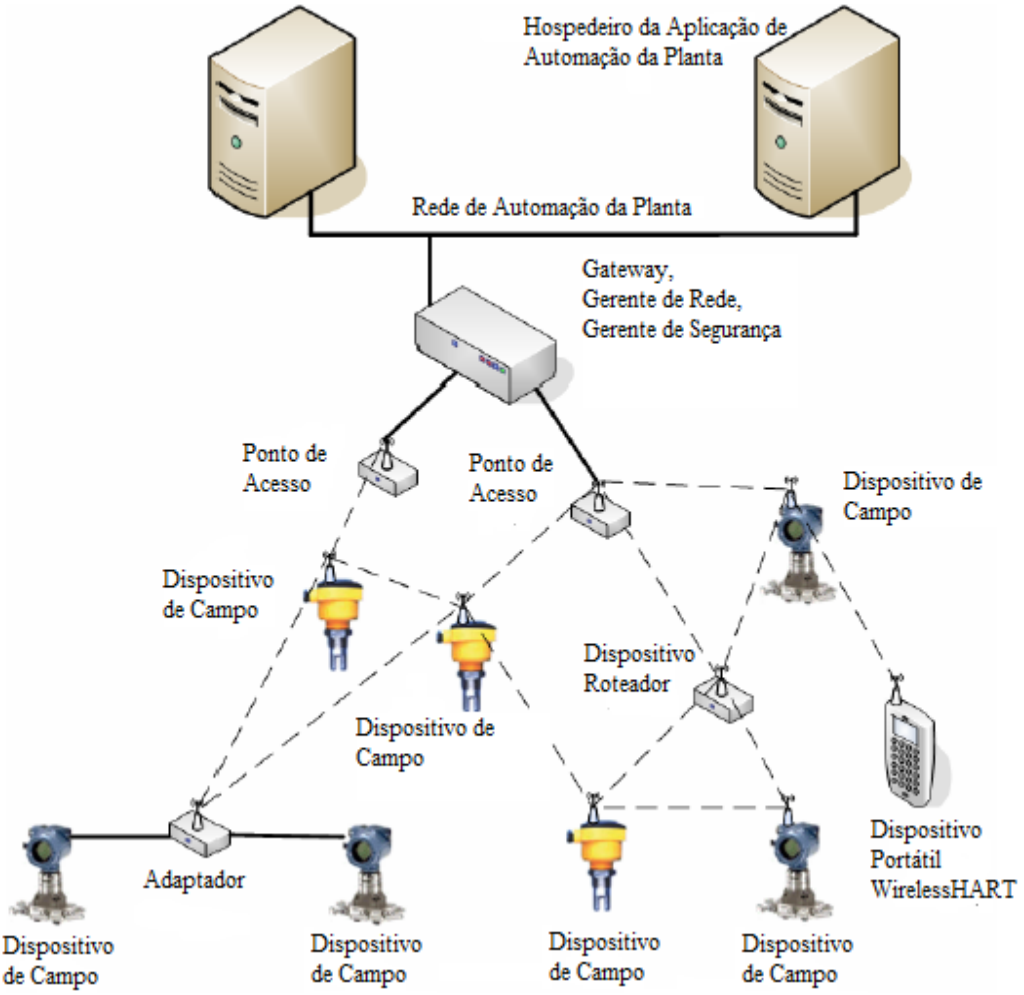
O protocolo HART versão 7 inclui um protocolo de comunicação novo, o protocolo WirelessHART, dando suporte a aplicações *wireless*. Assim como o protocolo HART cabeado, o Protocolo WirelessHART (PWH) tem como alvo sensores fixos e atuadores. Equipamentos de roteamento, equipamentos de rotação, como o secador de forno, e produção flexível também são mercados alvo. Precisa-se usar *wireless* para reduzir o custo de uma medição, acessar informação de diagnósticos avançados, e ter um melhor monitoramento de equipamentos. O PWH alavanca os protocolos existentes, como o HART, o IEEE-802.15.4, e o DDL/EDDL. Ele pode fazer tudo o que o HART cabeado pode fazer e muito mais. O PWH é uma tecnologia de rede segura operando na banda de rádio ISM de 2.4GHz. Utiliza rádios DSSS compatíveis com o IEEE 802.15.4 com *channel hopping* no conceito de pacote a pacote. A rede WirelessHART é compatível com uma larga variedade de dispositivos de diferentes fabricantes. A figura 1 ilustra os tipos de dispositivos de rede básicos, que incluem:

• Dispositivos de campo como dispositivos básicos desempenhando funções de sensoriamento de campo e funções de atuadores.
• Dispositivos de campo roteados, principalmente servindo como roteadores,
• Adaptador de dispositivo de campo conecta dispositivos HART cabeados a rede *MESH*,
• Dispositivo portátil WirelessHART transportado por usuários móveis é usado para unir um novo dispositivo de campo a uma rede WirelessHART já existente,
• Pontos de acesso conectam dispositivos de campo ao *gateway*,
• Um único *gateway* (pode ser redundante) age como *bridge* para as aplicações hospedeiras, conectando-as entre si ou a outra rede de comunicação de planta existente. Neste caso, no mesmo equipamento do *gateway* há o gerente de rede e o gerente de segurança, que gerencia e distribui chaves de encriptação de segurança, além de guardar a lista de dispositivos autorizados a unirem-se à rede WirelessHART, e,
• um *gerente de rede*, único (pode ser redundante), que constrói e mantém a rede *MESH*. Identifica os melhores caminhos e gerencia a distribuição de acesso de *slots* de

tempo (o WirelessHART divide cada segundo em *slots* de 10ms). O acesso de *slots* depende da taxa de atualização do valor do processo requerido e de outro acesso (registro de alarme – configuração - mudanças).

O hospedeiro da aplicação de automação da planta pode ser de gestão de ativos, por exemplo. A rede da automação da planta, que pode conectar vários hospedeiros de automação da planta.

Figura 1 – Exemplo de rede WirelessHART

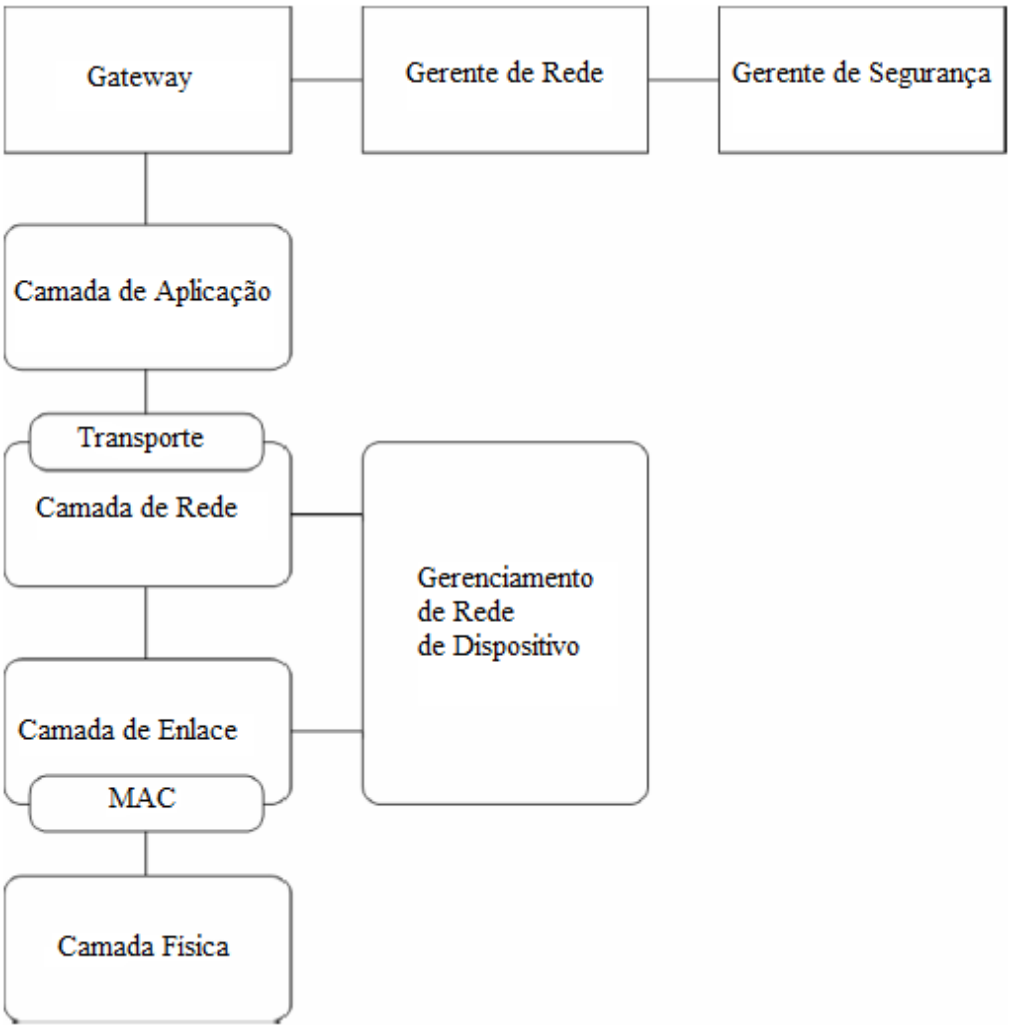


Fonte: (CHEN; NIXON; MOK, 2010, p. 24)

No PWH as comunicações são precisamente agendadas utilizando-se uma abordagem chamada de *Time Division Multiple Access* (TDMA). A grande maioria do tráfego é dirigida por um grafo de conexões. O agendamento é realizado por um Gerente de Rede centralizado que usa informações de roteamento de rede em combinação com os requisitos de comunicação que os dispositivos e as aplicações

fornecem. O agendamento está dividido em *slots* e é transferido a partir do Gerente de Rede para os outros dispositivos; os dispositivos recebem apenas os *slots* para os quais tenham requisitos de comunicação. O Gerente de Rede adapta continuamente o grafo geral da rede e o agendamento da rede a mudanças na topologia da rede e em demandas de comunicação. O relacionamento entre os componentes da pilha de comunicação WirelessHART pode ser visto na figura 2 (CHEN; NIXON; MOK, 2010, p. 24).

Figura 2 – Pilha de comunicação WirelessHART



Fonte: (CHEN; NIXON; MOK, 2010, p. 25)

A Fundação HART também oferece o Programa de Registro de Dispositivo HART para incluir dispositivos WirelessHART. O documento de procedimento de registro de dispositivos WirelessHART descreve os requisitos de teste e de registro para

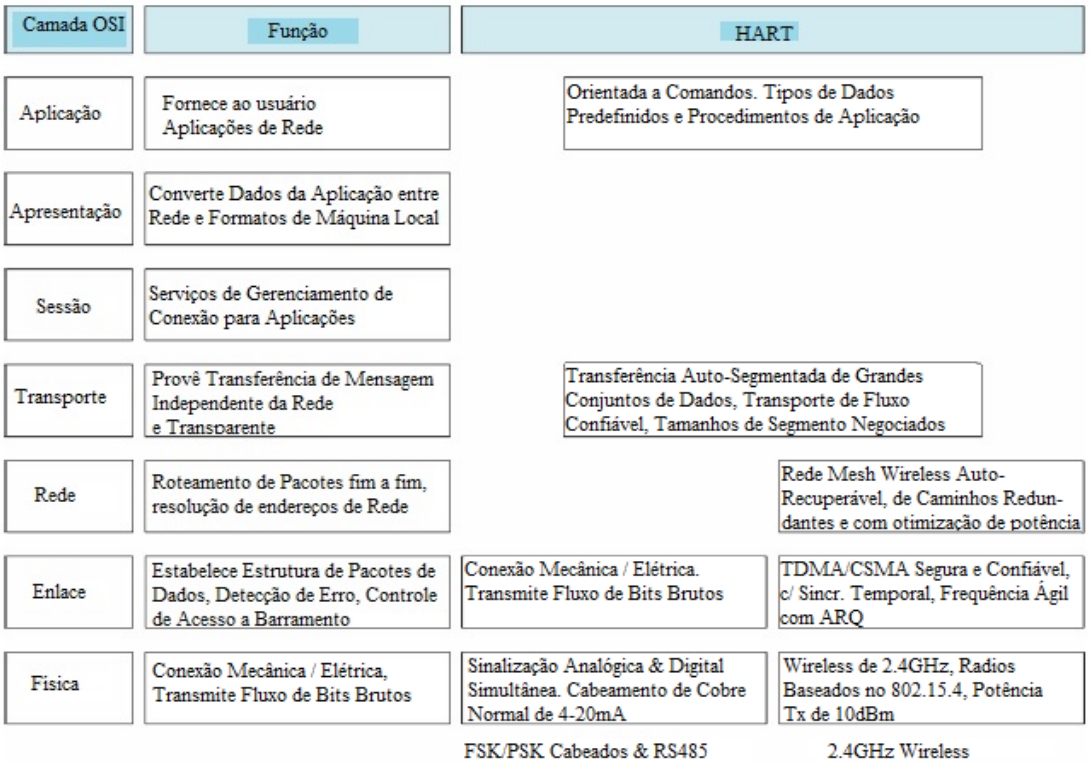
dispositivos WirelessHART. Teste e registro de um dispositivo garante a qualidade do dispositivo HART e a interoperabilidade de todos os dispositivos que reivindicam a conformidade HART.

Da mesma forma, os requisitos do procedimento de registro de dispositivo WirelessHART asseguram a interoperabilidade dos dispositivos *wireless* em um ambiente com múltiplos fabricantes. Além disso, eles garantem a conformidade desses dispositivos aos requisitos da especificação do protocolo de comunicação HART. Dispositivos que cumprem com os requisitos de registro estão autorizados a carregar a marca registrada HART (CHEN; NIXON; MOK, 2010, p. 25).

2.3 As Camadas

A figura 3 ilustra a arquitetura da pilha do PWH de acordo com o modelo de comunicação de 7 camadas OSI. Como visto na figura 3, a pilha do PWH inclui cinco camadas: a camada física, a camada de enlace, a camada de rede, a camada de transporte e a camada de aplicação. Além disso, um gerente de rede centralizado é responsável pelo agendamento do roteamento da comunicação da rede (CHEN; NIXON; MOK, 2010, p. 26).

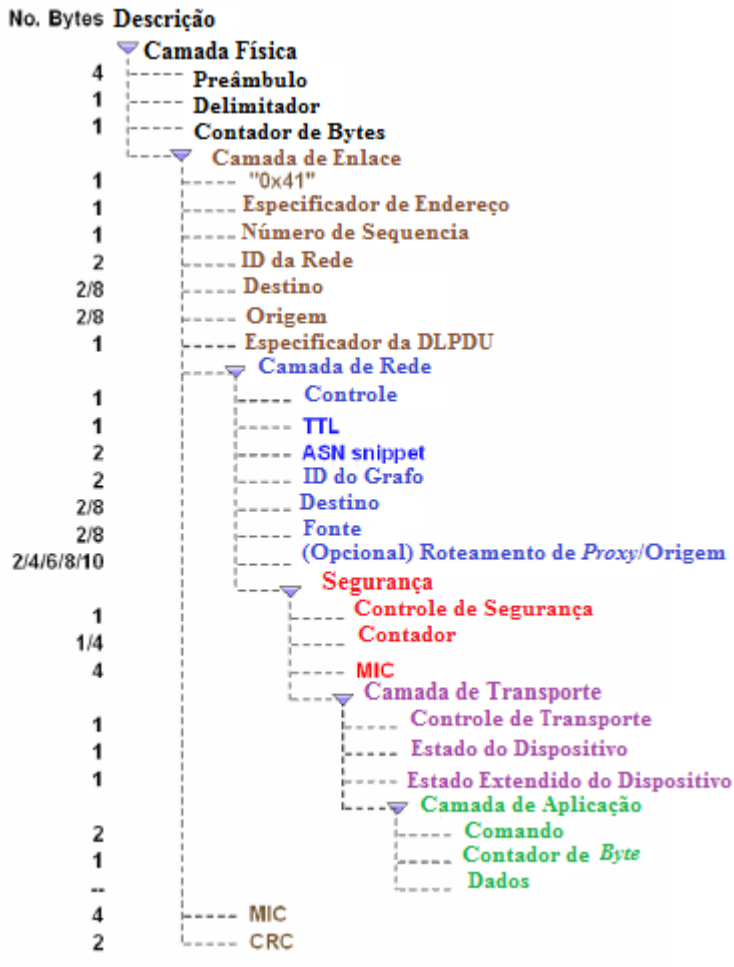
Figura 3 – Arquitetura da pilha do protocolo WirelessHART



Fonte: (CHEN; NIXON; MOK, 2010, p. 26)

A PDU (*Protocol Data Unit*) do PWH contém *bytes* em quantidade variável. Conforme figura 4, vê-se a PDU desde a camada física até a camada de aplicação.

Figura 4 – Resumo do formato da PDU



Fonte: (CHEN; NIXON; MOK, 2010, p. 48)

2.3.1 Camada Física

A camada física do WirelessHART é baseada majoritariamente na camada física do IEEE 802.15.4-2006 2.4GHz DSSS, e é um conjunto muito simplificado deste. Esta camada define características do rádio, tais como o método de sinalização, a intensidade do sinal, a sensibilidade do dispositivo, o nível de relação elétrico e físico entre um nó e um meio físico, antena, meio aéreo, nível de potência, tempo das variações de tensão,

taxas de dados físicos, máximas distâncias de transmissão, etc. Assim como o protocolo IEEE 802.15.4, o PWH opera na banda ISM 2400-2483.5MHz livre de licença com uma taxa de dados de até 250 kbits/s. Seus canais são numerados do 11 ao 26, com um intervalo de 5MHz entre dois canais adjacentes.

A camada física WirelessHART fornece serviços para a camada de enlace superior (HCF_SPEC -65 para mais detalhes). A camada física é responsável por enviar os bits através do meio de rede (CHEN; NIXON; MOK, 2010, p. 26; HART, 2007, p. 8).

Alguns parâmetros da camada física pertinentes ao trabalho são o RSL e o canal. O RSL (*receive signal level*) é o nível de sinal recebido estimado para um determinado pacote. O canal, no contexto do trabalho, representa o canal utilizado para a recepção de um determinado pacote.

2.3.2 Camada de Enlace

Uma característica distinta do PWH é o tempo sincronizado da camada de enlace. Define-se um *slot* de tempo de 10ms e utiliza-se a tecnologia TDMA para prover comunicações determinísticas e livres de colisão. O conceito de *superframe* é introduzido para agrupar uma sequencia de *slots* de tempo consecutivos. Um *superframe* é periódico, com o comprimento total dos *slots* de tempo como o período. Todos *superframes* em uma rede WirelessHART iniciam do ASN (*Absolute Slot Number*) 0, o momento em que a rede é primeiramente criada. Cada *superframe* então repete-se ao longo do tempo baseado no seu período. No PWH, uma comunicação em um *slot* de tempo é descrita por um vetor: {*frame id, index, type, src addr, dst addr, channel offset*}, onde *frame id* identifica o *superframe* específico; *index* é o índice do *slot* de tempo no *superframe*; *type* indica o tipo do slot de tempo (transmissão/recepção/inativo); *src addr* e *dst addr* são os endereços do dispositivo fonte e do dispositivo destino, respectivamente; *channel offset* provê o canal lógico a ser usado na comunicação.

Os parâmetros da camada de enlace pertinentes ao trabalho são: ASN, *src addr, dst addr, DLPDU Specifier, DLL MIC, DLL Authenticates, DLL Key Ysed, DLL Nonce Used, DLL ASN Used,* e *DLL Calc Mic*. Para realizar um ajuste fino no uso do canal, o

PWH introduz a ideia de *channel blacklisting*. Canais afetados por interferências consistentes podem ser incluídos na *blacklist*. Desta forma, o administrador de rede pode desabilitar totalmente o uso destes canais pela *blacklist*. Para dar suporte ao *channel hopping*, cada dispositivo mantem uma tabela de canais ativos. Portanto provê-se diversidade de canais e melhora-se a confiabilidade da comunicação.

A figura 5 descreve o projeto geral da camada de enlace, que consiste em seis módulos principais, como descrito a seguir.

Interfaces

A interface entre o Controle de Acesso ao Meio WirelessHARTe a camada física descreve as primitivas de serviço fornecidas pela camada física, e a interface entre o Controle de Acesso ao Meio WirelessHART e a camada de rede define as primitivas de serviço fornecidas pela camada de rede.

Temporizador

O temporizador é um módulo fundamental no PWH. Ele provê precisão temporal para assegurar a correta operação do sistema. Um desafio significativo faz-se em como projetar o módulo temporizador e manter os *slots* de tempo de 10ms em sincronização.

Tabelas de Comunicação

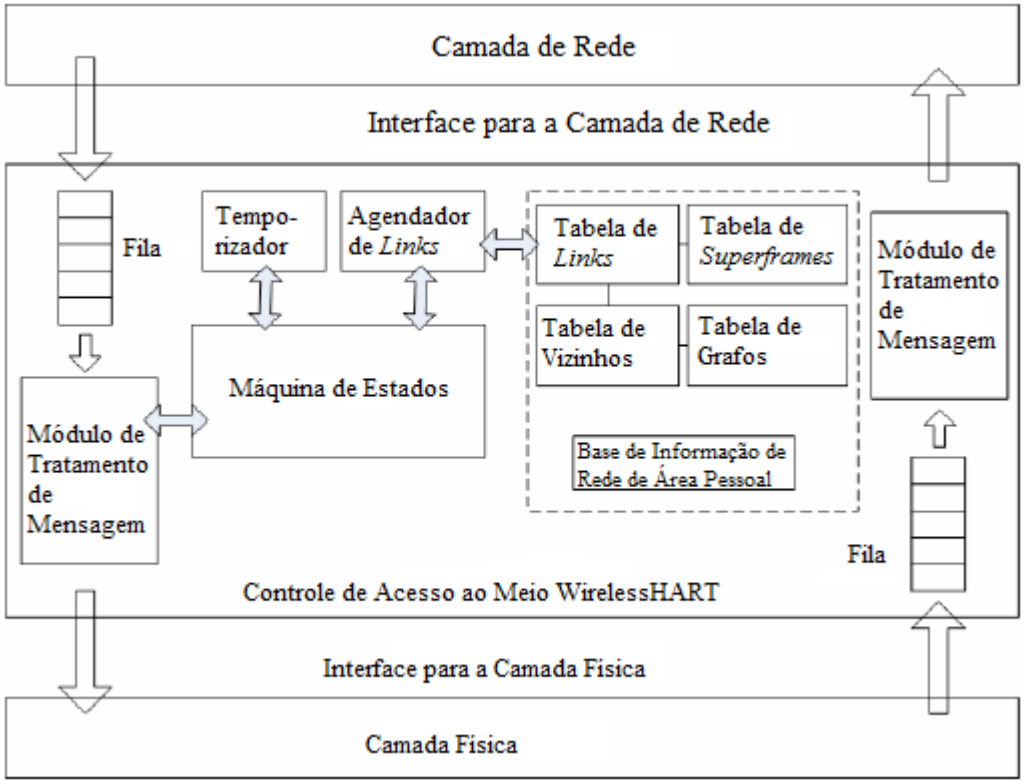
Cada dispositivo de rede mantém uma coleção de tabelas na camada de enlace. A tabela de *superframe* e a tabela de *link* armazenam as configurações de comunicação criadas pelo gerente de rede; a tabela de vizinhos é uma lista de nodos vizinhos os quais o dispositivo pode alcançar diretamente; e a tabela de grafo é usada para colaborar com a camada de rede e gravar informação de roteamento.

Agendador de *Links*

A função do agendador de *links* é determinar o próximo *slot* de tempo a ser usado baseado na agenda de comunicação na tabela de *superframe* e na tabela de *link.* O agendador é complicado por fatores como prioridades de comunicação, mudanças em

link e a ativação e desativação de *superframes*. Todo evento que pode afetar o agendamento de *link* implicará no reagendamento do *link*.

Figura 5 – Arquitetura da camada de enlace do protocolo WirelessHART



Fonte: (CHEN; NIXON; MOK, 2010, p. 28)

Módulo de Tratamento de Mensagem

O módulo de tratamento bufferiza os pacotes da camada de rede e da camada física separadamente.

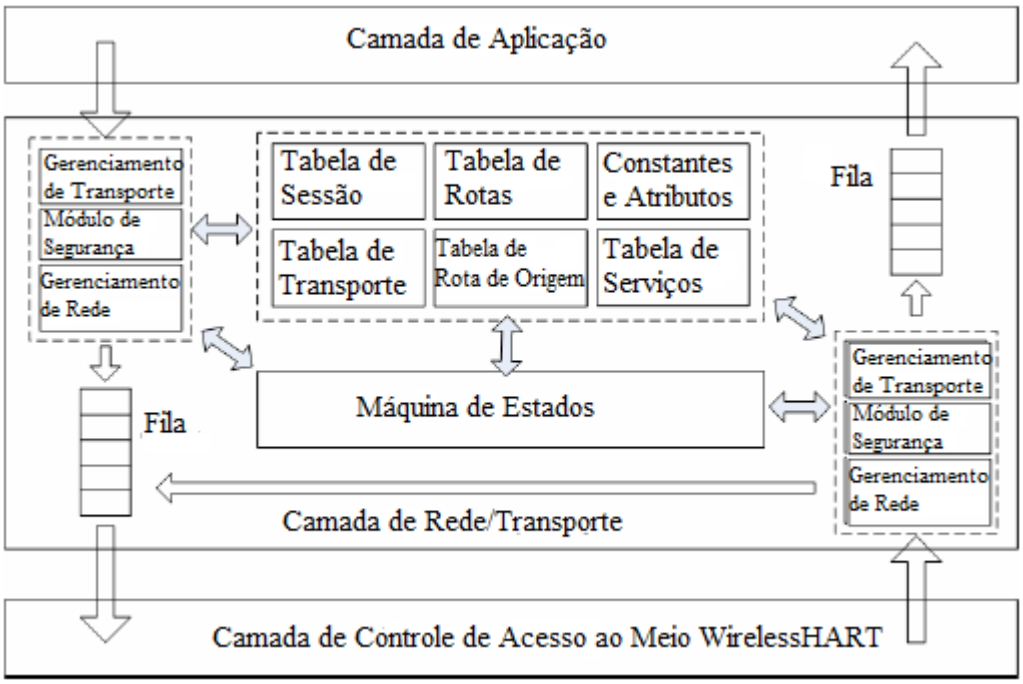
Máquina de Estados

A máquina de estados na camada de enlace consiste de três componentes primários: a máquina de estados TDMA, os motores XMIT e RECV. A máquina de estados TDMA é responsável por executar a comunicação em um *slot* e ajustar o relógio do temporizador. Os motores XMIT e RECV lidam com o *hardware* diretamente, que enviam e recebem um pacote através do tranceptor, respectivamente (CHEN; NIXON; MOK, 2010, p. 28).

2.3.3 Camada de Rede e Camada de Transporte

A camada de rede e a camada de transporte cooperam para prover comunicação fim-a-fim segura e confiável para os dispositivos de rede. A figura 6 descreve o projeto geral da camada de rede e de transporte.

Figura 6 – Arquitetura da camada de rede do PWH



Fonte: (CHEN; NIXON; MOK, 2010, p. 29)

Os elementos básicos de uma rede WirelessHART típica incluem: (1) Dispositivos de campo (DCs), que estão acoplados ao processo da planta, (2) Computador portátil habilitado para o WirelessHART para configurar dispositivos, executar diagnósticos, e realizar calibrações. (3) Um *gateway* que conecta aplicações hospedeiras com DCs, e (4) Um Gerente de rede que é responsável por configurar a rede, agendando e gerenciando a comunicação entre dispositivos WirelessHART.

Para operar adequadamente com a tecnologia de comunicação *mesh*, cada dispositivo WirelessHART deve ser capaz de encaminhar pacotes em nome de outros dispositivos. Há três protocolos de roteamento definidos no padrão WirelessHART:

Roteamento de Grafo

Um grafo é uma coleção de caminhos que conectam nodos de rede. Os caminhos em cada grafo são explicitamente criados e baixados para cada dispositivo de rede individual. Para enviar um pacote, o dispositivo fonte escreve um *ID* de grafo (determinado pelo destino) no cabeçalho de rede. Todos dispositivos de rede no caminho para o destino devem ser pré-configurados com informação de grafo que especifique os vizinhos para os quais os pacotes podem ser encaminhados.

Roteamento de Origem

É um roteamento que utiliza um grafo suplementar que objetiva diagnósticar a rede. Para enviar um pacote para seu destino, o dispositivo fonte inclui no cabeçalho uma lista ordenada de dispositivos através dos quais o pacote deve viajar. A medida que o pacote é roteado, cada dispositivo roteador utiliza o próximo endereço de dispositivo da lista para determinar o próximo *hop*. Isso é repetido até que o dispositivo destino seja alcançado.

Roteamento de *Superframe*

É um caso especial deroteamento de grafo. No roteamento de *superframe* pacotes são atribuídos a um *superframe*. Pacotes são instruídos a seguir a rota do *superframe* a partir da origem até o destino. Com roteamento de *superframe* o ID do Grafo é estabelecido para o ID do *Superframe*. Já que o pacote segue o *superframe*, não é necessário explicitamente configurar arestas do grafo (CHEN; NIXON; MOK, 2010, p. 30).

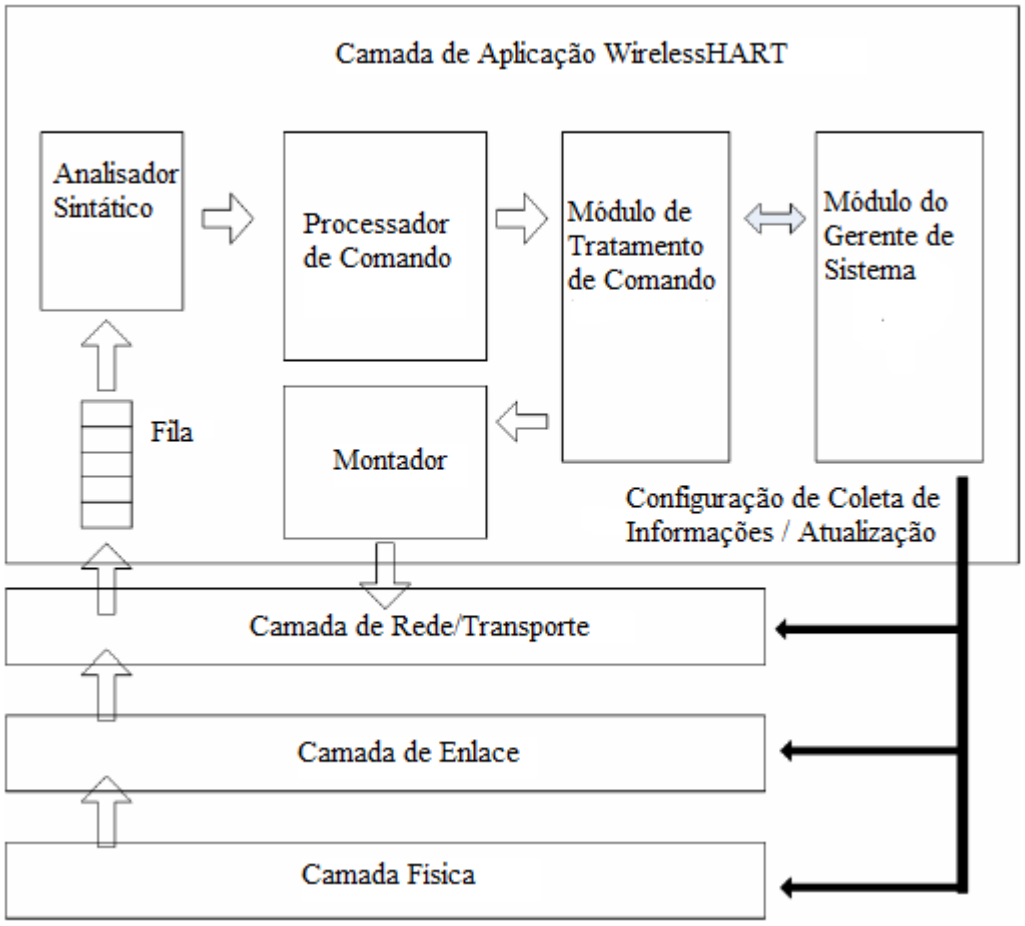
Os parâmetros da camada de rede e transporte pertinentes ao trabalho são: NL Ctl, *Graph* ID, *Dest*, *Src*, *Proxy*, *Src Route* 1, *Src Route* 2, TL Ctl, TL *Status* e TL *ExStatus*.

2.3.4 Camada de Aplicação e Arquitetura de Segurança

Na figura 7 está representado o projeto geral da camada de aplicação. Ela define comandos de vários dispositivos, respostas, tipos de dados e relatórios de estado. No PWH, a comunicação entre os dispositivos e o *gateway* é baseada em comandos e respostas. A camada de aplicação é responsável por investigar o conteúdo da

mensagem, extraindo o número do comando, executar o comando específico, e gerar respostas.

Figura 7 – Arquitetura da camada de aplicação do PWH



Fonte: (CHEN; NIXON; MOK, 2010, p. 30)

Arquitetura de Segurança

Uma rede WirelessHART é um sistema de rede seguro. Tanto a camada de Controle de Acesso ao Meio quanto a camada de rede proveem serviços de segurança. A camada de Controle de Acesso ao Meio provê integridade de dados *hop-to-hop* ao usar uma combinação de verificação de redundância cíclica, do inglês *cyclic redundancy check* (CRC) e um código de integridade de mensagem, do inglês *Message Integrity Code* (MIC). Apesar do CRC ter valor limitado, ele ainda é usado. Tanto quem envia quanto quem recebe usam o modo CCM* juntamente com AES-128 como cifra

de bloco subjacente para gerar e comparar o MIC. A camada de rede emprega várias chaves para prover confidencialidade e integridade de dados para conexões fim-a-fim.

Os parâmetros da camada de aplicação e da arquitetura de segurança pertinentes ao trabalho são: *Description, Packet Number, Date And Time, Elapsed Time, Packet Status*, PDU, *Priority*, ASN (0)), *Net* ID*, To, From, Advertise, Payload,* DLL MIC*, CRC, CRC Failure,*DLL *Key Used*, DLL *Calc Mic, PayloadHex, Payload,* SL Ctl, SL Cnt, SL MIC, SL *Decrypts,* SL *Key Used,* SL *Nonce Used,* SL *Calc Mic, Cipher Text* e *Clear Text.*

3 TESTES DE CONFORMIDADE COM A NORMA WIRELESSHART

3.1 Visão Geral

A interoperabilidade e o cumprimento das especificações do Protocolo HART são essenciais para o sucesso dos dispositivos. Deve haver monitoramento contínuo e integração de informação digital HART com sistemas de gerenciamento de ativos. Os testes de verificação de conformidade (TVCs) são realizados para verificar a conformidade de dispositivos com a norma WirelessHART. São testes fundamentais para garantir o funcionamento adequado dos dispositivos sob o Protocolo WirelessHART (PWH). Para auxiliar nos testes e no desenvolvimento de dispositivos HART, a HCF fornece ferramentas padronizadas para as empresas-membro e entidades parceiras. O Laboratório de Automação, Sistemas de Controle e Robótica (LASCAR), da UFRGS, é um dos parceiros e responsáveis por realizar TVCs.

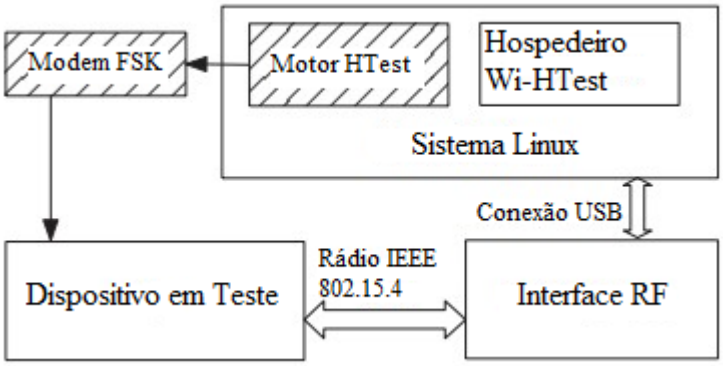
3.2 Topologia e Funcionamento dos Testes

Três equipamentos compõem a topologia de TVCs. O dispositivo em teste (DET), o Wi-HTest e o Wi-Analys. Para que a conformidade seja adequadamente testada, os três equipamentos devem estar dispostos de maneira que as antenas dos três estejam ao alcance entre si e que a distância entre eles não interfira nos resultados. O DET está conectado por cabo, utilizando a modulação *Frequency Shift Keying (*FSK*)*, ao Wi-HTest para comissionamento. Pode-se visualizar a topologia nas figuras 8, 9 e 10. Na figura 8 vê-se o Wi-HTest com seu módulo RF. Na figura 9 vê-se a arquitetura do sistema Wi-HTest com o DET. Na figura 10 está o Wi-Analys, responsável por capturar, filtrar, exibir e salvar em arquivos a comunicação de cada teste de conformidade.

Figura 8 – Equipamentos que compõem o Wi-HTest

Fonte: (HAN et al., 2009, p. 5)

Figura 9 – A arquitetura de alto nível do Wi-HTest



Fonte: (HAN et al., 2009, p. 5)

Figura 10 – Wi-Analys: captura tráfego da rede WirelessHART

Fonte: (HAN et al., 2009, p. 9)

Conforme figura 9, o DET é comissionado pelo Wi-HTest através do modem FSK (por cabo) para receber as configurações e chaves para a comunicação *wireless*. Após ser comissionado, o DET comunica-se por rádio com a interface RF do Wi-HTest ao longo do TVC. Enquanto isso, o Wi-Analys monitora a comunicação através de sua antena RF, apresenta os pacotes no *Software* Wi-Analys e salva um arquivo para cada teste executado.

O Wi-HTest é um componente essencial para os TVCs, porém, não o único. O Wi-HTest comunica-se com o DET enquanto ele estiver ativo na rede WirelessHART. O Wi-HTest não consegue coletar todas as informações de conformidade. Por exemplo, o Wi-HTest pode verificar a corretude do dispositivo nas camadas de rede, transporte e aplicação. Mas não é capaz de informar se o dispositivo está usando o algoritmo de retransmissão de mensagens correto e se os requisitos de tempo definidos na norma WirelessHART são respeitados. Um DET está em conformidade com a norma somente se todas suas transmissões *wireless* estiverem de acordo com a norma. Por esta razão, outros dois componentes de verificação de conformidade também são essenciais. Eles são o Wi-Analys e o *Post Processing Suite* (PPS). O Wi-HTest é quem gera todo o

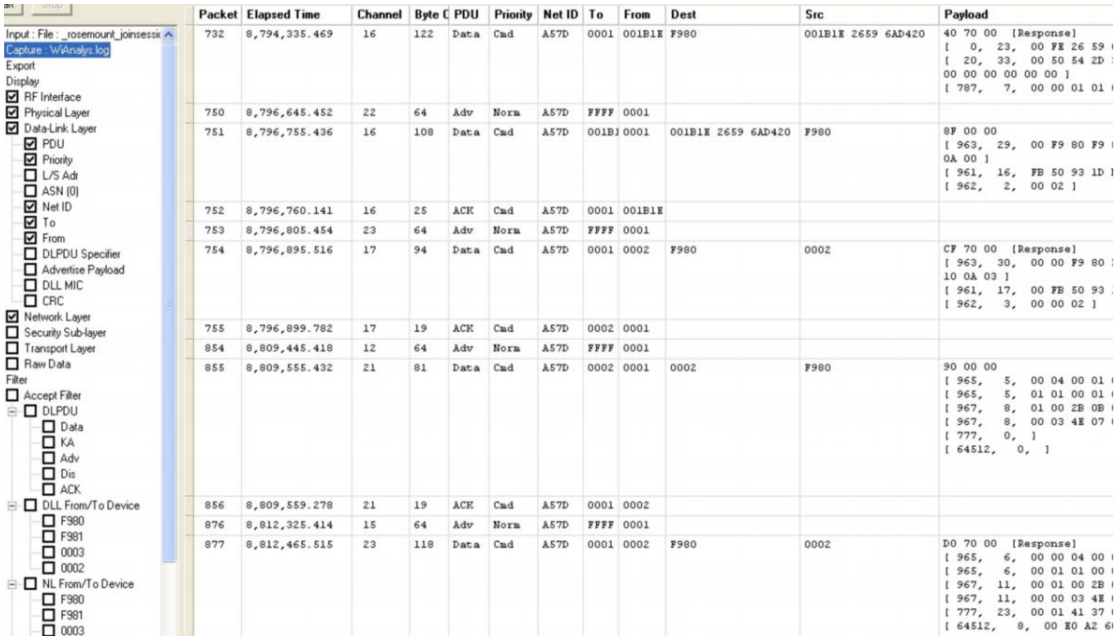
tráfego de comunicação em tempo real. O Wi-Analys captura e registra o tráfego *wireless* em arquivos de *log*. Os arquivos de *log* são pós-processados pelo PPS para verificar a conformidade de todas as mensagens com a norma. Os arquivos de *log* trazem dois benefícios. Eles servem como dados brutos para a verificação de conformidade e como evidência de defesa quando o Wi-HTest é responsável pela falha do DET.

O Wi-Analys é projetado para capturar todos os pacotes 802.15.4 na frequência 2.4 GHz, mas processa apenas aqueles provenientes de dispositivos WirelessHART. O receptor tem a capacidade de capturar dados em 16 canais WirelessHART simultaneamente e a uma velocidade de 100 mensagens por segundo. Como mostrado na figura 10, o Wi-Analys consiste em uma caixa receptora de rádio no centro e um *software suite* sendo executado em uma estação de trabalho. A caixa receptora está conectada a estação de trabalho via cabo USB. O *software suite* registra todas as mensagens WirelessHART capturadas em todos os canais. Ele apresenta as mensagens capturadas de maneira organizada, tanto *online* quanto reapresentando um arquivo de *log* criado anteriormente. Os campos das mensagens, desde a camada física até a camada de aplicação, são interpretados e apresentados em colunas. Além disso, o Wi-Analys decifra as mensagens de tal maneira que campo cifrados podem ser apresentados em formato texto. A figura 11a mostra uma captura de tela da interface visual do Wi-Analys. A figura 11a mostra um segmento parcial do DET juntando-se à rede. A figura 11b é uma ampliação de uma parte da figura 11a. Note-se que na Figura 11b o *channel hopping* está em efeito e é observado na coluna do canal. Também pode-se ver que as mensagens da camada de aplicação, formado por comandos e respostas HART, são apresentadas em formato texto. Como apresentado no lado esquerdo da figura 11a, o Wi-Analys tem um filtro que permite ao analista restringir a lista de mensagens ao especificar os parâmetros para cada campo. O Wi-Analys é um produto independente do HCF. Ele pode ser usado como uma ferramenta de monitoramento de rede WirelessHART em tempo real.

O PPS julga o êxito do TVC. Para cada caso de teste, um programa de pós processamento lê o arquivo de *log* e analisa-o. Dependendo do propósito de cada teste, ele checará a sequencia de mensagens transmitidas pelo DET, os *time points* de transmissão, a relação entre as mensagens, o conteúdo das mensagens, etc. Se tudo

estiver de acordo com a norma, o TVC passou. Se não, o lugar onde a norma está violada é reportado. O Wi-HTest, o Wi-Analys e o PPS constituem o ambiente de verificação de conformidade completo. Neste ambiente, um dispositivo passa por vários passos para ser certificado pelo HCF. Um teste completo é composto por um conjunto de casos de teste. O Wi-HTest executa cada caso de teste com o Wi-Analys capturando as mensagens ao longo de todo o período do teste. Enquanto o Wi-HTest pode informar se o dispositivo passou por certos casos de teste, um programa de pós processamento por caso irá analisar o arquivo de *log* correspondente para verificar se o dispositivo seguiu estritamente a norma durante o teste, especialmente se satisfez os requisitos de tempo (HAN et al., 2009, p. 9).

Figura 11a – Captura de tela de um segmento de sequencia de mensagens no caso de teste de *join* capturado pelo Wi-Analys



Fonte: (HAN et al., 2009, p. 10)

Figura 11b – Ampliação de parte captura de tela de um segmento de sequencia de mensagens no caso de teste de *join* capturado pelo Wi-Analys

Fonte: (HAN et al., 2009, p. 10)

3.3 Tabela de Testes

Há vários testes de verificação de conformidade WirelessHART isolados. Eles testam a conformidade de cada camada do protocolo, da física até a de aplicação. O Wi-HTest executa cada caso de teste e gera um arquivo de *log* com a ajuda do Wi-Analys e do PPS. Há diversos TVCs, como o teste de junção de dispositivo, o teste de gerenciamento de *superframe*, o teste de roteamento da rede, o teste de publicação de dados em rajada, e o teste de manutenção da rede.

Para desenvolver um DC compatível com o WirelessHART, uma variedade de testes *ad-hoc* informais e testes formais devem ser realizados. A HCF simplifica o esforço de teste fornecendo uma especificação de testes para os desenvolvedores. Todos os testes devem ser completados juntamente com o arquivo de *log* do teste antes da liberação do produto e de seu registro com a HCF. A especificação de testes fornece requisitos de teste claros e reduz o número de planos de teste que devem ser desenvolvidos pelo fabricante. Eles podem ser usados cedo no esforço de desenvolvimento para informalmente verificar funcionalidades durante a implementação e são uma parte útil do programa de teste em regressão enquanto o DC é melhorado. Além disso, a especificação de testes clarifica ambiguidadedes no protocolo e é a autoridade máxima na interpretação do PWH. O conjunto completo de testes consiste das seguintes cinco fases. Cada fase contém múltiplas descrições de teste e são projetadas para execução sequencial. As fases são: testes *bootstrap*, testes de

correspondência única, testes de correspondência múltipla, testes de seleção multi-canal e testes de estresse. Os testes *bootstrap* tentam auditar o conjunto de comandos implementados no DET através tanto da porta de manutenção quanto da conexão *wireless*. Eles também põem o DET em um estado de inicializado e testam seu processo de entrada em uma rede WirelessHART específica. Baseado na exitosa compleição de todos testes *bootstrap*, os testes de correspondência única focam em um canal RF lógico único e o DET interage diretamente com o gerente de rede e o *gateway*. Esta série de testes examina se um DC *wireless* requisita propriamente admissão para a rede *wireless*; aceita comandos que condicionam sua operação na rede *wireless*, incluindo comandos com uma execução adiada; e opera sincronizadamente com o dispositivo ao qual está conectado. Diferente dos testes de correspondência única, os testes de correspondência múltipla verificam se um dispositivo *wireless* interage adequadamente com múltiplos dispositivos aos quais esteja conectado, incluindo informação inferida sobre aqueles dispositivos a partir das mensagens recebidas. Testes de seleção multi-canal extendem os testes de correspondência múltipla. Eles asseguram que o dispositivo *wireless* seleciona propriamente, entre múltiplos canais em potencial, aquele cuja agenda permite operação. Por último, os testes de estresse combinam todos os testes anteriores em uma sequencia aleatória única que serve para examinar as operações de dispositivo contínuas em um ambiente de campo simulado. O principal propósito desta fase é de confirmar que o dispositivo irá interoperar confiavelmente com outro em um ambiente de mundo real. A HCF escreveu os TVCs para vários casos de teste em cada fase de teste. Os TVCs são pequenos e estritamente focados nas aplicações de teste. Eles são usados como entrada para alimentar o Wi-HTest para estabelecer os ambientes de teste, gerando pacotes de teste próprios e conduzindo os testes de conformidade. Tipicamente um TVC inclui duas partes: a configuração de teste e o corpo de teste. A seção de configuração de teste inicializa o gerente de rede e o *gateway*, configura o Wi-Analys e configura a interface RF e vários parâmetros de teste. Dispositivos virtuais necessários e agendas de comunicação correspondentes também são adicionadas ao suporte a testes de conformidade da camada de rede. O corpo de teste consiste em uma sequencia de pequenas etapas de teste. Cada etapa de teste gera ou manipula um pacote de dados WirelessHART ao chamar bibliotecas relacionadas implementadas no Wi-HTest. O corpo de teste então aguarda pela resposta do DET e verifica sua conformidade. O TVC

é escrito em C++. Para exemplos concretos, consulte o documento de especificação de testes (HAN et al., 2009, p. 9).

Na figura 12 pode-se ver a lista de todos os TVCs *wireless* disponíveis no Wi-HTest. São 117 TVCs no total, alguns levando mais de 24 horas para serem concluídos. Considerando-se isso, executou-se 17 TVCs para utilizar seus arquivos de *log* como entrada para a ferramenta implementada.

Figura 12 – Lista de todos os TVCs *wireless* disponíveis no Wi-HTest

TML100A	TML100B	TML100C	TML101	TML102A
TML102B	TML102C	TML201A	TML201B	TML202A
TML202B	TML202C	TML203A	TML203B	TML203C
TML203D	TML203E	TML203F	TML203G	TML203H
TML204A	TML204B	TML205A	TML205B	TML205C
TML205D	TML205E	TML205F	TML206A	TML206B
TML206C	TML206E	TML206D	TML206F	TML208A
TML208B	TML208C	TML208E	TML208D	TML208F
TML210A	TML210B	TML210C	TML211A	TML211B
TML212A	TML212B	TML212C	TML212D	TML213A
TML213B	TML213C	TML213D	TML213E	TML214A
TML214B	TML214C	TML214E	TML214D	TML214F
TML215A	TML215B	TML215C	TML215D	TML215E
TML215F	TML215G	TML215H	TML216A	TML216B
TML216C	TML216D	TML216E	TML216F	TML220A
TML222A	TML222B	TML222C	TML222D	TML222E
TML223	TML224A	TML224B	TML225A	TML225B
TML302A	TML303A	TML303B	TML303C	TML303D
TML303E	TML304A	TML304B	TML304C	TML304D
TML304E	TML305A	TML305B	TML305C	TML307A
TML307B	TML307C	TML307D	TML309	TML311A
TML311B	TML311C	TML312A	TML312B	TML312C
TML312D	TML314A	TML314B	TML401	TML402
TML501A	TML501B			

Fonte: do autor

3.4 Arquivos de *log* gerados pelo *Post Processing Suite* e pelo Wi-Analys

O arquivo de *log* mais completo gerado pelo *Post Processing Suite* é exibido na tela do Terminal Unix e segue na figura 13. O arquivo de *log* gerado pelo Wi-Analys exibe pacotes capturados pelo Wi-Analys referentes ao teste e também é exibido a seguir, na figura 13.

Figura 13 – Trecho de arquivo de *log* do *Post Processing Suite* do teste TML203A

```
---- Info :: DUT joined successfully


---- Info :: Write SF0, links and NM route...
Data Info: 04 7e 1f 20 00 00 00 04 f9 80 00 01 00 01 00 00 00 00 90 00 00
03 c5 05 00 00 65 01 00 03 c7 08 00 00 32 0a 00 01 02 00 03 c7 08 00 00
3c 0a 00 01 01 00 03 ce 05 00 f9 80 00 00 03 cb 03 00 01 01

write_data_to_port : fc 01 00 00 00 00 00 00 cd 8b 02 04 00 00 00 00 02
01 00 00 00 00 00 00 00 00 00 00 00 03 07 00 00 f4 01 ff ff 01 06 41 00
04 7e 1f 20 00 00 00 04 f9 80 00 01 00 01 10 78 5f 91 f9 89 30 a6 d1 b5
eb 99 c6 3c 4a e8 cd 91 a8 0c 94 7d 9f 04 eb 10 c4 90 bc 3b 0a 78 23 0a
38 e6 b0 38 3a 06 ec c6 55 f9 af ad f4 ff 03 04 6a


---- Info :: wait_data_from_port - Begin...
Get wanted msg from AP : post p_nwkreadReady : process_recvdata
re-constructed nonce: 00 00 00 00 02 00 00 00 00 00 00 00 04
deciphered apdu : 03 c5 06 00 00 00 65 01 0f 03 c7 0b 00 00 00 32 0a 00
01 02 00 00 3b 03 c7 0b 00 00 00 3c 0a 00 01 01 00 00 3a 03 ce 07 00 00
f9 80 00 00 06 03 cb 04 00 00 01 01

---- Info :: Write SF1 and links...
write_data_to_port : 03 c5 05 01 00 71 01 00
write_data_to_port : 03 c7 08 01 00 55 0a 00 04 02 00
Data Info: 00 7e 20 4c 00 00 00 04 f9 80 00 02 00 00 00 00 91 00 00 03 c5
05 01 00 71 01 00 03 c7 08 01 00 4b 0a 00 01 02 00 03 c7 08 01 00 55 0a
00 01 01 00

write_data_to_port : fc 01 00 00 00 00 00 00 cd 8b 02 04 00 00 00 00 02
01 00 00 00 00 00 00 00 00 00 00 00 03 07 00 00 f4 01 ff ff 00 07 31 00
00 7e 20 4c 00 00 00 04 f9 80 00 02 a5 28 b5 ce ae db f3 46 88 57 12 8a
2f 18 64 6b 05 72 81 89 99 ab 89 8e 65 0b 20 4e 58 e9 e8 98 3f 20 b0 a6
2d
```

Fonte: do autor

Figura 14 – Trecho de arquivo de *log* do Wi-Analys do teste TML203A

1	USB Devic	Descri	Packet	Date And T	Elaps	RSL	Packet	Chanr	Byte (	PDU	Priorit
2	119021	802.15	379	2015-07-24	####	-35	0x0000	21	62	Data	Cmd
3	119021	802.15	379	2015-07-24	####	-35	0x0000	21	62	Data	Cmd
4	119021	802.15	380	2015-07-24	####	-32	0x0000	12	44	Adv	Norm
5	119021	802.15	381	2015-07-24	####	-30	0x0000	17	44	Adv	Norm
6	119021	802.15	382	2015-07-24	####	-35	0x0000	17	62	Data	Cmd
7	119021	802.15	382	2015-07-24	####	-35	0x0000	17	62	Data	Cmd
8	119021	802.15	383	2015-07-24	####	-30	0x0000	23	44	Adv	Norm
9	119021	802.15	384	2015-07-24	####	-39	0x0000	17	16	KA	Norm
10	119021	802.15	385	2015-07-24	####	-37	0x0000	17	19	ACK	Norm
11	119021	802.15	386	2015-07-24	####	-29	0x0000	13	44	Adv	Norm
12	119021	802.15	387	2015-07-24	####	-38	0x0000	13	44	Adv	Norm
13	119021	802.15	388	2015-07-24	####	-88	0x0001	20	10		

Fonte: do autor

Como pode-se ver nas figuras 13 e 14, o arquivo de *log* do *Post Processing Suite* deixa claro para o analista de testes o que está acontecendo em cada trecho. Já o arquivo de *log* do Wi-Analys exibe apenas os pacotes referentes ao teste, sem informação explícita do que está acontecendo. Então, para analisar o comportamento do teste, o analista acaba tendo um grande trabalho de filtrar e buscar a informação que deseja em meio a uma grande quantidade de informações gerada pelo arquivo de *log* do Wi-Analys.

Conforme visto na seção 2.3, onde são especificadas as camadas e suas funções, também abordou-se parâmetros que são campos do arquivo de *log* do Wi-Analys: *USB Device Number, Description, Packet Number, Date And Time, Elapsed Time, RSL, Packet Status, Channel, Byte Count, PDU, Priority, L/S Adr, ASN (0)), Net ID, To, From, DLPDU Specifier, Advertise, Payload, DLL MIC, CRC, CRC Failure, DLL Authenticates, DLL Key Used, DLL Nonce Used, DLL ASN Used, DLL Calc Mic, NL Ctl, TTL, ASN Snippet, Graph ID, Dest, Src, Proxy, Src Route 1, Src Route 2, PayloadHex, Payload, SL Ctl, SL Cnt, SL MIC, SL Decrypts, SL Key Used, SL Nonce Used, SL Calc Mic, TL Ctl, TL Status, TL ExStatus, Cipher Text, e Clear Text.*

4 DESENVOLVIMENTO DA FERRAMENTA

4.1 Objetivo

Atualmente o analista de testes precisa verificar se um teste foi exitoso e se apresentou erros. Além disso, deve-se saber onde houve erros, falhas de autenticação, e comportamento inadequado do teste. Para isso, é necessário que o analista busque por essas informações no arquivo de *log* do *Post Proccessing Suite* e do Wi-Analys, que reúnem muita informação e acabam tornando esta tarefa demorada. Portanto, o objetivo desta ferramenta é facilitar e agilizar o trabalho do analista de testes gerando um arquivo de *log* final estilo planilha para isso.

4.2 Facilidades Propostas e Arquivo de *Log* Final Gerado

A ferramenta recebe como entrada o arquivo de *log* do *Post Processing Suite* e o arquivo de *log* do Wi-Analys, gerando como saída um arquivo de *log* final, que traz diversas facilidades:

Apresentar apenas os campos de interesse para analisar o teste. Os campos escolhidos foram: *FinalLine, DateAndTime, Type*, DLL ASN *Used, PayloadHexa*, CMD *Meaning, Payload*, *Application Data*, *Comment* e HTest *Line.*

FinalLine é um campo criado para indicar o número da linha no arquivo final. O arquivo de *log* final tem como base de ordenação a ordem dos pacotes no arquivo de *log* do Wi-Analys. Porém, há inserções de erros e não correspondências da segunda varredura de forma a sincronizar temporalmente os acontecimentos dos arquivos de *log* do *Post Processing Suite* e do arquivo de *log* do Wi-Analys. *DateAndTime* é um campo já presente no arquivo de *log* do Wi-Analys que representa uma informação temporal. Ele segue o formato:
“ANO-MÊS-DIA HORA:MINUTOS:SEGUNDOS:MILISSEGUNDOS”. Isso permite verificar a ordem temporal em que os pacotes foram capturados pelo Wi-Analys. Não há garantias de valores únicos em *DateAndTime*. *Type* é um campo já presente no arquivo de *log* do Wi-Analys. Ele pode ser de diversos tipos, como *advertisement*. Todavia, para

relevância de verificação de conformidade, apenas pacotes tipo ACK e *Data* são salvos no arquivo de *log* final. DLL ASN *Used* (*Data Link Layer Absolute Slot Number*) é um campo já presente no arquivo de *log* do Wi-Analys que representa o ASN da camada de enlace utilizado. É mais uma referência temporal para auxiliar na análise do teste. *PayloadHexa* é outro campo importado do arquivo de *log* do Wi-Analys e representa uma forma de *payload* expressa em *bytes* representados em hexadecimal. O primeiro número após a abertura de colchetes indica o número do comando HART ou WirelessHART. CMD *Meaning* é um campo criado para o arquivo de *log* final que informa o número e a função do comando HART ou WirelessHART presente no pacote. Além disso, informa-se em qual publicação detalhes sobre o mesmo podem ser encontrados. Exibir a função dos comandos e informar ao analista o que está ocorrendo no teste. Por isso, este é um campo de grande importância para acelerar e auxiliar na verificação de conformidade. *Payload* também é importado do arquivo de *log* do Wi-Analys. Este campo exibe uma pré-interpretação da camada de aplicação do pacote. Exibe o número do comando, o significado (muitas vezes de forma confusa), parâmetros do comando e variáveis como bc (*byte count*) e rc (*response code*). *Response code* é uma variável numérica muito importante pois pode informar indiretamente o que está ocorrendo de errado com o teste. *Response codes* são previstos na especificação do protocolo WirelessHART. *Application Data* é um campo criado pela ferramenta e representa os *bytes* da camada de aplicação (ver figura 5) do campo *ClearText* (presente no arquivo de *log* do Wi-Analys). Nele estão contidas todas informações da camada de aplicação. Com o desafio de encontrar correspondências de pacotes do arquivo de *log* do Wi-Analys com comentários do arquivo de *log* do *Post Processing Suite*, descobriu-se que o elemento comum é *Application Data*, que aparece nos dois arquivos de *log*. Este parâmetro está presente na camada de aplicação da norma WirelessHART (ver seção 2.3.4). Para extrair *Application Data* foi necessário o entendimento da pilha de camadas do protocolo WirelessHART (ver figura 4). A partir de então, implementou-se a solução, que considera diferentes possibilidades de contagem de *bytes*, visto que o número de *bytes* de cada camada oscila. Outra grande utilidade de *Application Data* é permitir importar o comentário (campo *Comment*) correspondente do arquivo de *log* do *Post Processing Suite*. Ver figura 4. Para o TVC TML102C, pode-se ver o mesmo *Application Data* nos arquivos

de *log* do PPS (ver figura 15) e do Wi-Analys (ver figura 16). Ele é salvo no arquivo de *log* final (ver figura 17).

Figura 15 – *Application Data* do TVC TML102C no arquivo de *log* do PPS

```
---- Info :: Start data logging in Wi-Analys
Data Info: 00 7e ad 90 00 00 f9 84 f9 80 00 01 00 00 00 00 90 00 00 00 80
18 74 6d 6c 31 30 32 63 00 00 00 00 00 00 00 00 00 31 2e 36 00 00 00 00
00
```

Fonte: do autor

Figura 16 – *Application Data* do TVC TML102C no arquivo de *log* do Wi-Analys

	AQ	AR	AS	AT	AU	AV	AW	AX	AY
1	SL Nonce	SL Calc Mi	TL Ctl	TL Status	TL ExStatu	Cipher Te	Clear Text		
2	00000000(	F1E77410	90	0	0	3E41882B			
3	00000000(	F1E77410	90	0	0	3E41882B	3E41882BCD8B84F9010037007E0		
4						2C41885E	1F40000F984F9800001F1E774109		
5						2C418872	00000008018746D6C31303263000		
6							0000000000000000312E360000000		
7	00000000(	F1E77410	90	0	0	3E418890(	0005202FD9304EB		

Fonte: do autor

Figura 17 – *Application Data* do TVC TML102C no arquivo de *log* final

H2 : 00 80 18 74 6d 6c 31 30 32 63 00 00 00 00 00 00 00 00 00 31 2e 36 00 00 00 00 00

	A	B	C	D	E	F	G	H	I	J
1	F	DateAr	Typ	DLL AS	Paylc	CMD M	Payloa	Application Data	Comment	HTest Lir
2	2	2016-05-1	Data	0000000:	90 00 00	- - - - - - - -		00 80 18 74 6d 6c 31	---- Info :: Start data logg	190

Fonte: do autor

Comment é um campo criado pela ferramenta, que apresenta comentário(s) presente(s) no arquivo de *log* do *Post Processing Suite* que correspondem a linha em questão do arquivo de *log* do Wi-Analys. Esse campo é fundamental para o analista entender quais etapas foram executadas em dado momento do teste e se houve comportamento inadequado. Pode haver mais de uma correspondência para o mesmo *Application Data*. Neste caso, mais de um comentário aparecerá no campo *Comment*. Os comentários são separados por "//". As figuras 15 e 17 apresentam o mesmo comentário para o mesmo *Application Data*, mostrando a correta importação do

comentário para o arquivo de *log* final. Para que o analista tenha clara a sequência temporal dos erros em relação a execução do teste, recomenda-se ordenar as linhas pelo campo HTest *Line*. HTest *Line* é um campo criado para o arquivo de *log* final e informa o número da linha onde foi encontrada a primeira correspondência. Pode-se ver todos os campos do arquivo de *log* final nas figuras 18 e 19.

Figura 18 – Campos do Arquivo de *log* Final

1	FinalLine	DateAndTime	Type	DLL ASN Used	PayloadHexa	CMD Meaning
2	2	2016-05-16 20:09:42.921	Data	000000022B000000	90 00 00 *[12	- - - - - - - - - -
3	3	2016-05-16 20:09:42.921	Data	000000022B000000	90 00 00 *[12	128 - Veja o Doc
4	4	2016-05-16 20:09:43.937	Data	2,9E+18	90 00 00 *[12	- - - - - - - - - -
5	5	2016-05-16 20:09:43.937	Data	2,9E+18	90 00 00 *[12	128 - Veja o Doc
6	6	2016-05-16 20:09:54.031	Data	6,82E+18	91 00 00 *[12	- - - - - - - - - -
7	7	2016-05-16 20:09:54.031	Data	6,82E+18	91 00 00 *[12	128 - Veja o Doc
8	8	2016-05-16 20:09:55.046	Data	00000006E700000(	91 00 00 *[12	- - - - - - - - - -
9	9	2016-05-16 20:09:55.046	Data	00000006E700000(	91 00 00 *[12	128 - Veja o Doc
10	10	2016-05-16 20:10:03.125	Data	0000000A0F00000(	92 00 00 *[13	138 - Write Trav

Fonte: do autor

Figura 19 - Campos do Arquivo de *log* Final

Payload	Application Data	Comment	HTest Line
	00 80 18 74 6d 6c 3	---- Info :: St	194
[WaStartLogSession c	00 80 18 74 6d 6c 3	---- Info :: St	194
	00 80 18 74 6d 6c 3	---- Info :: St	194
[WaStartLogSession c	00 80 18 74 6d 6c 3	---- Info :: St	194
	00 80 18 74 6d 6c 3	---- Info :: St	194
[WaStartLogSession c	00 80 18 74 6d 6c 3	---- Info :: St	194
	00 80 18 74 6d 6c 3	---- Info :: St	194
[WaStartLogSession c	00 80 18 74 6d 6c 3	---- Info :: St	194
[WaSetIntTestParam (	00 8a 14 64 65 76 5	---- Info :: Se	205

Fonte: do autor

Conforme vê-se nas figuras 18 e 19, pode-se ordenar o arquivo pela ordem em que aparecem no arquivo de *log* final através de *FinalLine*, como também pela ordem em que aparecem no arquivo de *log* do *Post Processing Suite*, através do campo HTest *Line*. As principais funcionalidades do *log* final gerado pela ferramenta são:

• Apresentar correspondências e não-correspondências de conteúdo do arquivo de *log* do *Post Processing Suite* para o arquivo de *log* do Wi-Analys e vice-versa. Isso pode ajudar o analista de testes a entender o comportamento do teste.

• Apresentar os erros gerados pelo arquivo de *log* do *Post Processing Suite*. Este arquivo de *log* já apresenta uma interpretação do andamento do teste, com comentários sobre o que está acontecendo, inclusive reportando erros que aparecem ao longo do teste. Estes erros em forma de comentário são incluídos no arquivo final.

• Apresentar o número de erros encontrados no arquivo de *log* do *Post Processing Suite*. O número de erros aparece na última linha do arquivo final. Essa informação pode ajudar o analista a entender se o teste ocorreu como esperado e quantos problemas enfrentou.

• Apresentar correspondência temporal entre os conteúdos dos arquivos de *log*. O arquivo final é composto por linhas extraídas dos arquivos de *log* do Wi-Analys e do *Post Processing Suite*. Procura-se manter uma correspondência temporal entre essas informações pelo posicionamento das linhas entre si e através dos campos *FinalLine* e HTest *Line*.

4.3 Algoritmo

O arquivo de *log* final é montado visando-se poder ordená-lo de duas maneiras diferentes. A partir do arquivo de *log* do Wi-Analys e a partir do arquivo de *log* do *Post Processing Suite*, isto é, a partir dos campos *FinalLine* e HTest*Line* (ver figuras 18 e 19). Portanto, para construí-lo, são necessárias duas varreduras. Na primeira varredura, insere-se no arquivo de *log* final os pacotes tipo ACK e *Data* do arquivo de *log* do Wi-Analys. Para cada pacote tipo *Data*, extrai-se do campo *ClearText, Application Data* (ver seção anterior). Utiliza-se *Apllication Data* para buscar correspondências no arquivo de *log* do *Post Processing Suite*. Cada correspondência encontrada é adicionada ao arquivo de correspondências. Cada correspondência é uma linha com dois campos, onde o primeiro é o número da linha no arquivo de *log* final e o segundo é o número da

linha no arquivo de *log* do *Post Processing Suite*. Uma linha do arquivo de *log* do Wi-Analys pode encontrar diversas correspondências no arquivo de *log* do *Post Processing Suite*. Quando não há correspondência de conteúdo, isso é informado no campo "*Comment*". Além disso, quando há falhas de autenticação ou de decriptação, elas são informadas nos campos "*Comment*" ou "*PayloadHexa*". O número do comando HART ou WirelessHART é extraído do campo *Payload* e utilizado para buscar seu significado no arquivo "comandos.txt". Então, é informado o significado do comando e em qual especificação maiores detalhes do mesmo podem ser encontrados.

A segunda varredura tem como base o arquivo de *log* do *Post Processing Suite*. Varre-se o arquivo a partir do início em busca do comentário que indica o início da comunicação *wireless*. Então, busca-se comentários iniciados por "----", que podem relatar acontecimentos normais do teste ou erros. Busca-se, através do número da linha do comentário, se há correspondências do mesmo no arquivo de correspondências. Caso não haja, devemos inserir um elemento no arquivo de correspondências referente a esse comentário. Para isso, percorre-se o arquivo de correspondências novamente a partir do início. Quando for encontrado o primeiro elemento cujo número da linha de seu comentário for maior que o do elemento a ser inserido, esta será a posição de inserção. A linha do arquivo final do elemento a ser inserido será igual a linha do elemento escolhido já presente no arquivo. Já o elemento cuja posição foi escolhida para inserção do novo, deverá passar para a posição seguinte. Ele e todos os elementos seguintes são incrementados em uma unidade no campo "linha no arquivo final". Feito isso, o comentário é inserido no arquivo final na posição prevista no arquivo de correspondências. Para facilitar o diagnóstico do analista, os comentários da segunda varredura tem seus campos não utilizados preenchidos com "#####". Por último, a ferramenta insere na última linha do arquivo final, o número de erros encontrados na segunda varredura. O arquivo de correspondências contem elementos que representam as correspondências da primeira varredura e as não correspondências da segunda varredura. Cada elemento (linha) contém dois campos. O primeiro representa o número da linha no arquivo final. O segundo indica o número da linha no arquivo de *log* do *Post Processing Suite*, como vê-se na figura 20. O arquivo de *log* final deve estar nomeado da seguinte maneira: "NomeWiAnalys_Final.txt", onde "NomeWiAnalys" será o segundo argumento passado para a ferramenta.

Figura 20 – Exemplo de trecho de arquivo de correspondências

```
8       194
8       200
9       205
10      205
11      209
12      209
259     245
```

Fonte: do autor

Figura 21 – Trecho de arquivo de comandos

```
768 Write Join Key
769 Rad Join Status
770 Request Active Advertising
771 Force Join Mode
772 Read Join Mode Configuration
773 Write Network Id
774 Read Network Id
775 Write Network Tag
```

Fonte: do autor

4.4 Condições de Funcionamento

Para que a ferramenta funcione de maneira adequada, deve-se respeitar os seguintes critérios:

- Ambiente Windows XP ou Windows mais recente.

- O arquivo “comandos.txt”, o arquivo de *log* do *Post Processing Suite* e o arquivo de *log* do Wi-Analys devem estar presentes na mesma pasta do executável da ferramen-ta.

- Na linha de comando, deve-se navegar até o diretório onde localiza-se o executável através do comando “cd”. Então, deve-se executar o executável da ferramenta.

- Para rodar a ferramenta deverá ser usado o seguinte comando:

CriadorMescladorPlanilhas.exe nomepostprocesssuite nomewianalys

“nomepostprocesssuite” é o nome do arquivo de *log* do PPS, com a extensão “.txt”.

"nomewianalys" é o nome do arquivo de *log* do Wi-Analys, com a extensão ".txt".

5 TESTES DA FERRAMENTA

Testou-se a ferramenta para diversos casos de *script* de teste WirelessHART. Porém, problemas ainda em diagnóstico na operação do *kit* de testes WirelessHART resultaram em erros e comportamentos inadequados para a maioria dos *scripts* de teste executados. Grande parte dos testes da tabela 1, portanto, são *scripts* de testes que enfrentaram algum tipo de anormalidade e não tiveram sua execução como esperado. A ferramenta, de qualquer maneira, foi executada sobre todos os testes de conformidade da tabela 1.

Na Tabela 1 são apresentados alguns testes de conformidade utilizados para testar a ferramenta. No total há 117 testes de conformidade diferentes. Veja mais em TMLTestDescription_r1.0.pdf.

Tabela 1 – Alguns Testes de Conformidade WirelessHART

Teste de Conformidade	*Função*
TML100B	Auditar via Conexão *Wireless*
TML102C	Retentativas de Junção
TML203A	Aleatoriamente adicionar e remover *Superframes* inativos
TML203F	Manipular *Superframe* de Dispositivo Portátil
TML205A	Gerenciamento de Tabela de Rota
TML205D	Gerenciamento de Rota de Grafo e de Arestas
TML210B	Habilidade do DET de inferior colisões, recuo e retentativas de transmissão
TML303C	Roteamento de Grafo – *Broadcast* não reconhecido
TML304D	Roteamento de *Proxy* – *proxy* não é um vizinho

TML311C	Testar precedência & prioridade do pacote, rejeitar de acordo com limite de prioridade

5.1 Caso de Teste de Conformidade com Sucesso

O caso escolhido é o *script* de teste de conformidade com a norma WirelessHART denominado TML203A. O script TML203 tem como denominação "*Superframe Table Management*". É responsável por testar o gerenciamento da tabela de *superframes*.

O TML203A aleatoriamente adiciona e deleta *superframes* inativos em dois passos:

1. Adiciona *superframes* inativos e um número aleatório de *links* para um dos *superframes* existentes até que a tabela de *superframes* ou a tabela de *links* estejam cheias.

2. Antes que as tabelas estejam cheias, remove um *superframe* toda vez que houver algumas interações, e lê de volta a lista de *superframes* e a lista de *links*.

Figura 22 – Trecho de arquivo de *log* do *Post Processing Suite*

```
---- Info :: wait_data_from_port - Begin...
Get wanted msg from AP : post p_nwkreadReady : process_recvdata
---- Info :: Received write join keys response
re-constructed nonce: 00 00 00 00 01 00 00 00 00 00 00 00 04
deciphered apdu : 03 c1 11 00 16 31 e9 10 0b cd 6d 1d 0b a0 41 54 4b 7f
ef 47 03 c2 03 00 00 04 03 c3 1e 00 00 f9 80 f9 80 00 00 01 00 00 00 00
97 aa 04 79 f7 ca fe 95 62 80 6e 91 28 c2 08 c5 06
```

Fonte: do autor

Figura 23 – Trecho de arquivo de *log* do Wi-Analys

1	Cipher Te	Clear Text
535	5E4188B7	
536	134188B7	5E4188B7CD8B010004003F002042B12301F980000400014F76A63
537	2C4188C1	BCF700003C111001631E9100BCD6D1D0BA041544B7FEF4703C2
538	2C4188DF	0300000403C31E0000F980F98000000100000000097AA0479F7CAF
539	51418812	E9562806E9128C208C5069B05E52916EC

Fonte: do autor

Figura 24 – Trecho de arquivo de *log* final

42	42	2015-07-2	Data	0000001F:	8F 00 00	961 Write	[WriteNe	eb 8f 00 0(	SEM CORR		714
43	43	#####	#####	#####	#####	#####	#####	#####	---- Info ::		751
44	44	#####	#####	#####	#####	#####	#####	#####	---- Info ::		757
45	45	2015-07-2	ACK	0000001F:		--------					714
46	46	2015-07-2	Data	0000001F:	CF 70 00	[961 Write	[WriteNe	03 c1 11 0(	---- Info ::		766
47	47	2015-07-2	Data	0		--------			auth failu		766

Fonte: do autor

Figura 25 – Trecho de arquivo de *log* final ordenado por HTest*Line*

1	FinalL	DateAr	Type	DLL AS	Payloa	CMD M	Payloa	Applica	Comme	HTest L	
55	55	2015-07-2	ACK	0	auth failu	--------					780
56	56	2015-07-2	Data	00000020:	91 00 00	965 Write	[WriteSu	03 c5 05 01	---- Info ::		785
57	57	2015-07-2	ACK	00000020:		--------					785
58	58	2015-07-2	Data	2,1E+19	D1 50 00	965 Write	[WriteSu	03 c5 06 0(	---- Info ::		795
59	59	2015-07-2	ACK	2,1E+19		--------					795
60	60	2015-07-2	Data	2,16E+19	92 00 00	963 Write	[WriteSe	03 c3 1d 0(	---- Info ::		799
61	61	2015-07-2	ACK	2,16E+19		--------					799

Fonte: do autor

Pode-se ver nas figuras 22 e 23 a correspondência de *Application Data*. A mesma correspondência aparece no arquivo de *log* final, como visto na figura 24. Pode-se ver ainda na figura 24, falhas de autenticação, não correspondências a partir do Wi-Analys (informe “SEM CORRESPONDÊNCIAS”) e não correspondências a partir do PPS (informe “#####”).

Pode-se observar na figura 24 que a ordenação está feita pelo campo *FinalLine*, que tem como base a ordenação do arquivo de *log* do Wi-Analys. Por isso, neste caso, o campo *DateAndTime* também estará ordenado, com algumas não correspondências intercaladas. Porém, é possível ordenar o arquivo de *log* final diretamente por esse campo, presente na segunda coluna. Por outro lado, como vê-se na figura 25, também é possível ordenar o arquivo de *log* final pelo campo HTest*Line*. Essa terceira forma de ordenação pode ser interessante para o analista entender como sucedeu-se o teste de verificação de conformidade tendo como base de ordenação o arquivo de *log* do *Post Processing Suite*, embora esse arquivo de *log* não tenha base temporal.

Como vê-se no campo *Comment*, o penúltimo, na figura 25, quando o teste é exitoso há poucas não correspondências e erros no arquivo de *log* final.

5.2 Caso de Teste de Conformidade com Problemas

Escolheu-se o TVC com a norma WirelessHART denominado TML202C. O TVC TML202 tem como denominação "*Neighbor Table Management*". É responsável por testar o gerenciamento da tabela de vizinhos.

O TML203C testa a lista de *links* e a tabela de vizinhos:

1. Lê a saúde do vizinho.
2. Adiciona um *link*, então lê a lista de *links*.
3. Adiciona e deleta *links* para vários vizinhos até que a tabela de *links* e a tabela de vizinhos estejam cheias.

Figura 26 – Trecho de arquivo de *log* final do TVC TML202C com problemas

150	2016-05-1	Data	0000003A	C9 50 00	[967 Write	[WriteLir	03 c7 01 4	SEM CORR	0
151	2016-05-1	ACK	0000003A		- - - - - - - -				0
152	2016-05-1	Data	0000003B	96 00 00	135 - Veja	[WaSetTe	00 87 10 4	SEM CORR	0
153	2016-05-1	Data	0000003B	96 00 00	135 - Veja	[WaSetTe	00 87 10 4	SEM CORR	0
154	2016-05-1	Data	0000003C	96 00 00	135 - Veja	[WaSetTe	00 87 10 4	SEM CORR	0
155	2016-05-1	Data	0000003C	96 00 00	135 - Veja	[WaSetTe	00 87 10 4	SEM CORR	0
156	2016-05-1	Data	0000003D	97 00 00	135 - Veja	[WaSetTe	00 87 10 3	SEM CORR	0
157	2016-05-1	Data	0000003D	97 00 00	135 - Veja	[WaSetTe	00 87 10 3	SEM CORR	0
158	2016-05-1	Data	0000003E	97 00 00	135 - Veja	[WaSetTe	00 87 10 3	SEM CORR	0
159	2016-05-1	Data	0000003E	97 00 00	135 - Veja	[WaSetTe	00 87 10 3	SEM CORR	0
160	2016-05-1	Data	4,12E+03	8A 00 00	960 Discor	[Disconn	03 c0 01 0	SEM CORR	0

Fonte: do autor

Como pode ser visto nas figura 26, quando o teste apresenta problemas, há muitas não-correspondências indicando comportamento indesejado do TVC.

6 CONCLUSÃO E TRABALHOS FUTUROS

A possibilidade de criar um único arquivo de *log*, unindo apenas informações pertinentes à análise dos testes e com sincronização temporal aceleraria e facilitaria a análise dos testes. Este trabalho propôs e implementou uma ferramenta que seleciona, une, organiza e pré-analisa algumas informações do teste, gerando um arquivo de *log* resultante único.

Para que a conformidade seja adequadamente testada, o dispositivo em teste, o Wi-HTest e o Wi-Analys devem estar dispostos de maneira que as antenas dos três estejam ao alcance entre si e que a distância entre eles não interfira nos resultados.

Como vê-se no capítulo 5, utilizou-se dois casos de teste para demonstrar os resultados do funcionamento da ferramenta: um caso de teste de verificação de conformidade com sucesso; outro com teste de verificação de conformidade com problemas. A primeira varredura exibe correspondências e não-correspondencias a partir de pacotes do arquivo de *log* do Wi-Analys para comentários do arquivo de *log* do *Post Processing Suite*. Já a segunda varredura exibe correspondências e não-correspondencias a partir de comentários do arquivo de *log* do *Post Processing Suite* para pacotes do arquivo de *log* do Wi-Analys.

Para o caso de teste em que houve sucesso, notou-se a predominância de correspondências sobre não-correspondências tanto para a primeira, quanto para a segunda varredura. Torna-se fácil e intuitivo para o analista de testes de conformidade compreender que os arquivos de *log* do Wi-Analys e do *Post Processing Suite* apresentam informações compatíveis sobre o funcionamento do teste.

No caso de teste em que houve problemas, há predominância de não-correspondências sobre correspondências, principalmente na primeira varredura. Desta forma, ao percorrer o arquivo de *log* resultante e deparar-se com esse padrão, torna-se fácil e rápido para o analista perceber que se trata de um teste com problemas. Além disso, o campo *Comment* e o campo *Command* presentes no arquivo de *log* final ajudam o analista a compreender quais foram os erros e as suas causas.

Ao final, os resultados mostraram que a ferramenta funcionou adequadamente para o objetivo proposto e tornou-se útil para a análise dos testes de verificação de conformidade de dispositivos com a norma WirelessHART.

Em um trabalho futuro seria interessante que se exibisse os *response codes* e seus significados. Os erros e as não-correspondências servem como indicadores de possíveis problemas nos testes de conformidade. Outros indicadores, como mensagens de etapas concluídas (por exemplo, "*Test Stimulus Complete*"), podem ser buscados e utilizados como parâmetros para diagnóstico dos testes.

REFERÊNCIAS

AUTOMATION. **Emerson Proves Advancements in EDDL (Electronic Device Description Language) Technology**. 2005. Disponível em: <http://www.automation.com/content/emerson-proves-advancements-in-eddl-electronic-device-description-language-technology>. Acesso em 9 de junho de 2016.

CHEN, Deji; NIXON, Mark; MOK, Aloysius. **WirelessHART**: Real-Time Mesh Network for Industrial Automation. Nova Iorque: Springer US, 2010.

HAN, Song et al. **Wi-HTest: Compliance Test Tool for Real-Time WirelessHART™ Mesh Network Devices**. Austin, TX: [s.n.], 2009.

HART Communication Foundation. **2.4GHz DSSS O-QPSK Physical Layer Specification** (**HCF_SPEC-065).** Revisão 1.0. Austin, TX: [s.n.], 2007.

HARTCOMM. **O que é o HART?**. 2014a. Disponível em: <http://pt.hartcomm.org/hcp/tech/aboutprotocol/aboutprotocol_what.html>. Acesso em 9 de junho de 2016.

HARTCOMM. **How HART Works**. 2014b. Disponível em: <http://webcache.googleusercontent.com/search?q=cache:fwY_xX5sMTIJ:en.hartcomm.org/hcp/tech/aboutprotocol/aboutprotocol_how.html+&cd=1&hl=pt-BR&ct=clnk&gl=br>. Acesso em 9 de junho de 2016.

HARTCOMM. **Visão Geral da Tecnologia WirelessHART**. 2014c. Disponível em: <http://pt.hartcomm.org/hcp/tech/wihart/wireless_overview.html>. Acesso em 9 de junho de 2016.

WIKIPEDIA. **Highway Addressable Remote Transducer Protocol**. 2016. Disponível em: <https://en.wikipedia.org/wiki/Highway_Addressable_Remote_Transducer_Protocol>. Acesso em 9 de junho de 2016.

ANEXO – TRABALHO DE GRADUAÇÃO 1

Este anexo consiste na primeira parte do trabalho de graduação em Engenharia de Computação entregue no segundo semestre do ano de 2015.

PROPOSTA DE FERRAMENTA PARA TESTES DE VERIFICAÇÃO DE CONFORMIDADE DE DISPOSITIVOS COM O PROTOCOLO WIRELESSHART

Alexandre Bento Leal, Sérgio Luís Cechin, João Cesar Netto

Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)

Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{ableal, cechin, netto}@inf.ufrgs.br

Abstract. *This paper presents the WirelessHART protocol and its need for compliance with WirelessHART devices, in a matter of temporal response and correctness according to HCF specifications. For this purpose, conformance tests are performed. However, the current procedure has limitations and it can be improved. Based on this, this paper presents the proposal of a software that automates the compliance testing procedure. In the performed study, it was concluded that the reports' synchronism, fields' filters and interpretation of communication are features that will streamline test procedure and it will be offered by the tool.*

Resumo. *Este artigo apresenta o protocolo WirelessHART e a necessidade de conformidade dos dispositivos WirelessHART com o mesmo, em questão de resposta temporal e corretude, segundo especificações da HCF. Para isso, são realizados testes de conformidade. Contudo, o procedimento atual tem limitações e pode ser aprimorado. Baseando-se nisso, este artigo traz a proposta de uma ferramenta que automatize o procedimento de testes de conformidade. No estudo realizado, concluiu-se que o sincronismo de relatórios, os filtros de campos e a interpretação da comunicação, são funcionalidades que irão agilizar o procedimento e serão oferecidas pela ferramenta.*

1 1. INTRODUÇÃO

Em ambientes industriais, a tecnologia wireless apresenta facilidade de manutenção, implantação, configuração e seu custo está decrescendo. Existe uma necessidade crescente por novas medições de processos. A tecnologia wireless é confiável, segura e econômica, sendo uma boa solução para transmitir valores de medição aos sistemas de controle. Além disso, melhorias nos processos, mudanças físicas em instalações e exigências de níveis de segurança favorecem a escolha da tecnologia sem fio, HCF (2014b).

Atualmente, o padrão HART e o padrão WirelessHART estão sendo utilizados em larga escala em ambientes industriais. O padrão WirelessHART foi lançado em setembro de 2007 e tornou-se um padrão IEC em abril de 2010 (IEC 62591). Para garantir a conformidade com o protocolo de comunicação HART e a adequação a rígidos requerimentos temporais, todos os dispositivos WirelessHART devem ser completamente testados e registrados com a HCF (*HART Communication Foundation*), Han (2009).

Por operar na faixa de frequência de 2,4 GHz (isenta de licença), o padrão WirelessHART traz como vantagem permitir a coexistência com outras redes sem fio baseadas em tecnologias diferentes. Além disso, para evitar interferências, o protocolo utiliza um mecanismo de saltos entre canais de frequência. O WirelessHART é baseado em TDMA (*Time Division Multiple Access*), que é um mecanismo simples e oferece garantias temporais, dividindo ciclicamente o espaço em fatias temporais de transmissão (*slots*), atribuídos para cada usuário, Kurose e Ross (2006).

Com o propósito de garantir conformidade dos produtos HART (e WirelessHART) ao padrão, a HCF garante que todos os dispositivos HART sejam completamente testados e registrados. Para isso, foram desenvolvidas especificações de testes e as correspondentes ferramentas para realiza-los.

O Wi-HTest e o Wi-Analys, desenvolvidos pela HCF, fornecem um ambiente para a verificação de conformidade de dispositivos WirelessHART (DUT – *Device Under Test*), Han (2009). A comunicação sem fio, conforme uma rede WirelessHART, ocorre entre

o Wi-HTest e o DUT. O Wi-Analys captura o tráfego proveniente dessa comunicação, pacote a pacote, e salva-o em relatórios. O Wi-HTest também salva relatórios referentes a essa comunicação. Esses relatórios são utilizados para encontrar as causas de eventuais erros ocorridos em testes que apresentarem falhas.

O procedimento atual de verificação de conformidade e depuração de dispositivos HART ainda é predominantemente manual, no que diz respeito à detecção de erros. Isso faz com que, não somente o tempo de avaliação seja longo, mas acarreta dificuldades na identificação dos pontos de falhas. Quando um teste é realizado com sucesso, esta informação é facilmente encontrada no relatório do Wi-HTest. Contudo, quando o teste falha, o usuário tem o trabalho de analisar os logs do Wi-HTest e do Wi-Analys, junto com a norma do WirelessHART, para descobrir a causa e o local dos erros, o que pode tomar muito tempo.

Este trabalho apresenta a proposta de uma ferramenta para auxiliar a interpretação dos resultados dos testes de verificação de conformidade de dispositivos, com o protocolo WirelessHART. Essa ferramenta propõe-se a poupar o operador de algumas etapas demoradas e tediosas de análise e trazer uma pré-interpretação dos erros dos testes. A ferramenta trará funcionalidades como sincronismo, filtros de campos e uma interpretação da comunicação. Isso acelerará a interpretação de erros e por consequência a verificação de conformidade de dispositivos WirelessHART.

2 2. PROTOCOLO WIRELESSHART

A rede WirelessHART é centralizada, e seu maior componente é o Gerente de Rede, responsável pela formação e escalonamento da rede, Rech e Netto (2012). O protocolo WirelessHART utiliza o TDMA, que é utilizado no acesso aos canais do WirelessHART. O TDMA permite que a comunicação entre dispositivos ocorra em slots de tempo distintos, Petersen e Carlsen (2009).

Cada rede WirelessHART compreende três elementos principais, conforme pode ser visto na figura 1:

Os *Field Devices* (dispositivos de campo) são diretamente conectados aos processos ou a outros dispositivos de campo pela rede sem fio. Podem ser dispositivos HART conectados a um adaptador sem fio (*wireless adapter*). São equipamentos que, por exemplo, coletam medições de temperatura de um fluído e as enviam para o gerente de rede.

O *Gateway* realiza a comunicação entre a rede WirelessHART e os aplicativos host conectados a um *backbone* de alta velocidade ou a outra rede de comunicação existente na planta.

O Gerente de rede (*network manager*) é o elemento central da rede. Tem como funções a configuração da rede, o agendamento de comunicações entre os DUTs, o gerenciamento das rotas de mensagens e a manutenção do correto funcionamento da rede. O Gerente de rede pode estar integrado ao Gateway, ao aplicativo host ou ao controlador de automação de processos, HART (2014a).

Em geral, o Gerente de Segurança (*security manager)* está integrado ao gerente de rede. Ele é responsável por gerar, armazenar e gerenciar as chaves de segurança da rede. Essas chaves são utilizadas pelos dispositivos de campo para juntarem-se à rede. Além disso, é utilizada uma chave diferente para cada sessão de comunicação.

A Aplicação Hospedeira (*Application Host*) e o Controlador de Automação de Processo (*Process Automation Controller*) são utilizados para monitorar, gerenciar e controlar

a operação e os equipamentos da planta industrial. Comunicam-se com os dispositivos de rede através do *Gateway*.

O Computador Portátil Sem Fio (*Wireless Handheld*) é um dispositivo portátil utilizado nas tarefas de calibragem dos transmissores dos dispositivos de campo assim como outras tarefas gerenciais que os operadores tenham que desenvolver no trabalho de campo.

O Ponto de Acesso (*Access Point*) faz a conexão entre os dispositivos de campo e os outros elementos da rede. Ele costuma estar integrado ao Gateway ou ao Gerente de rede.

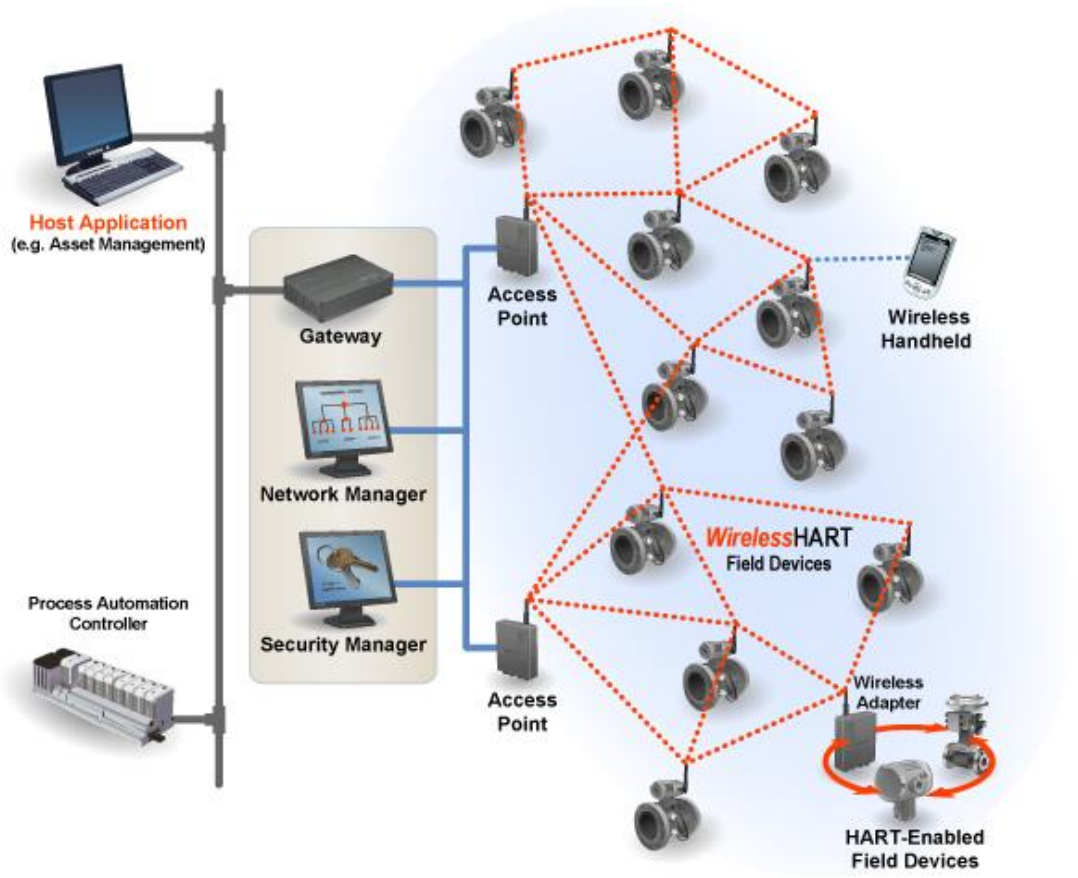


Figura 1 – A topologia de uma rede WirelessHART, HART (2014a).

Na rede *mesh*, um dispositivo de campo pode agir como um roteador de mensagens de dispositivos vizinhos. Ou seja, um dispositivo de campo não necessariamente deve estar diretamente conectado ao Gateway. Basta que envie a mensagem ao dispositivo de campo que estiver mais perto. Essa característica aumenta o alcance da rede e cria caminhos de comunicação redundantes, elevando a confiabilidade.

O Gerente de Rede estabelece os caminhos redundantes de acordo com características como confiabilidade, latência e eficácia. Com frequência, mensagens em sequência seguem por rotas redundantes para que essas mantenham-se abertas e desobstruídas. Dessa forma,

caso uma mensagem não consiga atingir o destino por uma rota, ela é redirecionada de maneira automática para uma rota redundante, sem prejuízo à comunicação.

Outra grande vantagem da rede *mesh* consiste na facilitação da inclusão ou da mudança de área de operação dos dispositivos de campo. Um dispositivo de campo pode se comunicar com qualquer outro, a menos que estejam fora da área de alcance mútua, HART (2014a).

3 3. PROCEDIMENTOS DE TESTES DE VERIFICAÇÃO DE CONFORMIDADE

O modelo de arquitetura para os testes é formado pelo Wi-HTest, o DUT (device under test) e o Wi-Analys:

Wi-HTest: Corresponde ao gateway e ao Gerente de Rede. Conta com um sistema operacional Unix e uma aplicação que permite a execução dos testes do padrão WirelessHART. Exibe e grava um arquivo log que corresponde a toda a comunicação HART e WirelessHART que passa pelo Wi-HTest, durante um determinado teste.

DUT: É o dispositivo de campo que será verificado. Ele recebe as configurações da rede sem fio (etapa de comissionamento) do Wi-HTest, via cabo RS232-C. A seguir, iniciam os testes, realizados pela comunicação, através da rede WirelessHART.

Wi-Analys: É o dispositivo que captura a comunicação da rede WirelessHART, registra os pacotes transferidos em arquivos de log e a apresenta para o usuário.

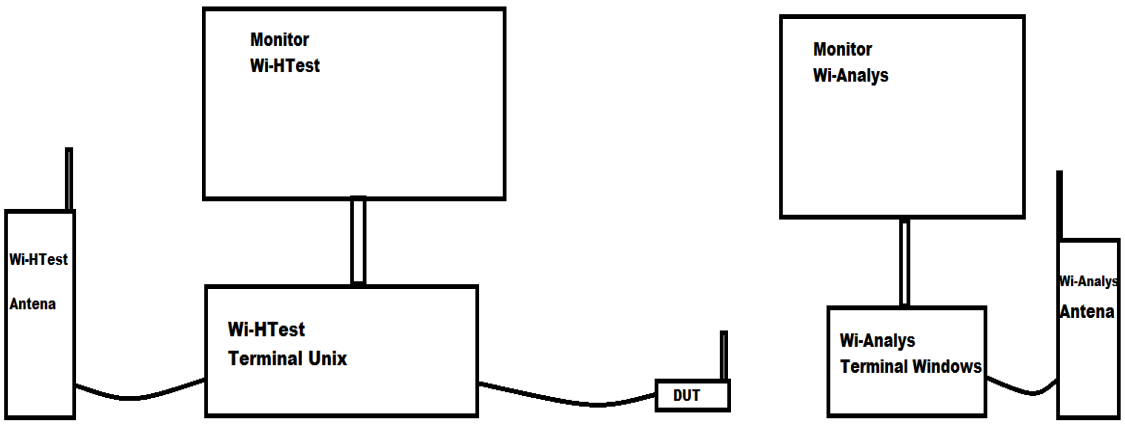


Figura 2 – Estrutura para testes de verificação de conformidade de dispositivos ao protocolo WirelessHART.

A HCF fornece uma especificação com diversos testes. Um teste corresponde a uma sequência de mensagens trocadas entre o Wi-HTest e o DUT.

Os testes de verificação de conformidade são automatizados e organizados em grupos. Cada grupo de comandos é usado para verificar uma camada do protocolo WirelessHART. Quando os resultados de um teste não apresentam falha de conformidade, as informações relativas a este teste são facilmente encontradas no log do Wi-HTest. Contudo, quando um teste falha, o operador precisa consultar os relatórios do Wi-HTest, do Wi-Analys e a norma do protocolo, para identificar causa e o local da falha. Nesse caso, o *response code* (um código de estado que informa tipos de erros) desse ser localizado e seu valor comparado com aquele da especificação da HCF, que explica o seu significado. Essa etapa é importante para identificar em qual parte do firmware do dispositivo está localizada a falha. Além disso, é importante que o operador entenda o teste como um todo. É preciso identificar os comandos e verificar se a resposta correspondente está de acordo com a norma. Todas essas tarefas devem ser realizadas manualmente por parte do operador, o que atrasa o procedimento de verificação de conformidade.

4 4. A SOLUÇÃO

Para que se possa analisar de forma adequada possíveis erros na comunicação, a ferramenta deve apresentá-los de forma a facilitar a sua interpretação. Uma das maiores dificuldades consiste em identificar a correspondência entre os trechos da comunicação presentes no relatório do Wi-HTest com os pacotes do relatório do Wi-Analys. Para facilitar essa tarefa, a ferramenta apresentará um sincronismo temporal entre esses relatórios. A ferramenta oferecerá filtros de campos, pois o relatório do Wi-Analys contém uma lista com um número elevado de campos para cada pacote identificado e, no momento da análise, em geral, são poucos os campos de interesse. Ainda, o operador perde algum tempo com interpretações dos relatórios que podem ser automatizadas. Para poupar o trabalho do operador, uma pré-interpretação da comunicação também será exibida.

4.1 4.1. O sincronismo

Para permitir que o operador possa analisar as informações recebidas pelos dois equipamentos (Wi-HTest e Wi-Analys), visando a identificação da causa de eventuais erros, é necessário sincronizar os relatórios (*logs*) fornecidos por cada um desses equipamentos.

Portanto, durante a sincronização, a ferramenta deverá apresentar a correspondência entre o relatório fornecido pelo Wi-HTest e os registros dos pacotes capturados pelo Wi-Analys.

O relatório fornecido pelo Wi-HTest filtra e interpreta a comunicação, fornecendo apenas um conjunto de bytes de interesse para o operador. Sáo removidas as redundâncias presentes quando se analisa vários pacotes trafegados pelo rede. Esse período de tempo que corresponde aos pacotes capturados, analisados e que geraram esse registro será chamado de uma “etapa” de comunicação.

O relatório do Wi-Analys, por sua vez, exibe todos os pacotes referentes à comunicação.

Portanto, devido a forma como os equipamentos geram seus relatórios, para cada trecho do relatório do Wi-HTest (etapa) haverá um conjunto correspondente de pacotes registrados no relatório do Wi-Analys.

.

O principal desafio da ferramenta é identificar o início e o término dessas etapas, a partir da análise dos relatórios fornecidos pelos dois equipamentos.

Para realizar essa sincronização, a ferramenta deverá comparar, entre outras informações, as marcas de tempo (*time stamps*) fornecidas no relatório de cada um dos dois equipamentos. Essa tarefa envolve a comparação das marcas de tempo na busca do melhor ajuste das informação de tempo obtidas dos relatórios, uma vez que é pouco provável encontrar eventos cujas marcas de tempo sejam exatamente iguais.

Finalmente, depois de sincronizados os relatórios, o operador poderá realizar, facilmente, a análise dos resultados, identificando os erros de comunicação.

4.2 4.2 Os filtros de campos

A ferramenta exibirá o relatório do Wi-Analys em uma janela. Os filtros de campos serão aplicados sobre esse relatório (arquivo no formato XML). Neste arquivo as informações de interesse são apresentadas de forma clara, em uma tabela onde cada coluna corresponde a um campo. No arquivo existe, para cada pacote, campos como *payload*, ASN (*Absolute Slot Number*), endereços de destinatário e fonte, prioridade, canal, número do pacote, CRC (*Cyclic redundancy check*), chave de segurança utilizada, entre outros. Com essa quantidade de informação, o operador demora para encontrar a informação de interesse. Dessa forma, com os filtros, será possível, por exemplo, exibir apenas pacotes que tenham como endereço de fonte o Gerente de Rede e endereço de destino o *Gateway*. Essa filtragem poderá ser realizada sobre todos os campos.

Além disso, como a lista de campos é extensa, será possível ocultar a exibição de quantos campos se queira. Na figura 3 é apresentada a forma como os campos são apresentados pelo Wi-Analys.

	Descriptic	Packet Number	Channel	Byte Cour	PDU	Priority	ASN (0))	Net ID	To	From	CRC
2	802.15.4-I	15626	21	62	Data	Cmd	2B	8BCD	wianalys	1	E4E7
3	802.15.4-I	15626	21	62	Data	Cmd	2B	8BCD	wianalys	1	E4E7
4	802.15.4-I	15630	22	64	Adv	Cmd	0	1	FFFF	1	FFE9
5	802.15.4-I	15631	14	10							
6	802.15.4-I	15632	23	64	Adv	Cmd	10	1	FFFF	1	FFEB

Figura 3 – Exemplo de relatório do Wi-Analys.

4.3 4.3 A interpretação da comunicação

A ferramenta interpretação da comunicação analisará os dados fornecidos nos relatórios do Wi-HTest e Wi-Analys e esses resultados serão exibidos em uma janela especialmente projetada para apresentar esses resulados..

Os resultados poderão ser apresentados de uma dentre duas formas posíveis: sobre trechos do teste e sobre o teste como um todo.

Para otimizar ao máximo o trabalho de análise do operador, a ferramenta utilizará informações de comunicação disponíveis na norma do protocolo WirelessHART, de testes específicos; do relatório do Wi-HTest e do relatório do Wi-Analys, especialmente de *response codes*. Com essas informações será possível determinar o que um teste está avaliando, quais comandos do protocolo estão sendo utilizados, quais falhas estão ocorrendo, suas causas e locais de ocorrência. Isso facilitará e acelerará o trabalho de verificação de falhas dos testes.

Em caso de falha no firmware do DUT, para se encontrar seu local de ocorrência precisa-se analisar os *response codes* juntamente com a norma do protocolo WirelessHART.

Por exemplo, se a ferramenta identificar uma falha, isso será informado nos resultados. Essas mensagens de falha podem ser apresentadas ao lado dos trechos onde foi identificada a falha, nos relatórios do Wi-HTest e do Wi-Analys. Caso não sejam encontradas falhas, a informação de sucesso no teste será exibida.

5 5. CRONOGRAMA

Tarefa/Mês	Jul	Ago	Set	Out	Nov
(1)					
(2)					
(3)					

Tabela 1. Cronograma para a segunda parte do trabalho de graduação.

(1) Projeto da ferramenta.

(2) Implementação da ferramenta.

(3) Escrita da monografia.

6 6. CONCLUSÕES

Um dispositivo deve passar por diversos testes para que esteja em conformidade com o padrão WirelessHART. É necessário que esse comporte-se de maneira esperada a uma sequência de commandos do protocolo.

Embora os testes de conformidade sejam automatizados, em caso de erros, existe a necessidade de interação do operador de testes. Isso implica em tarefas tediosas e demoradas, que poderiam ser automatizadas para acelerar os procedimentos de teste para certificação.

Sabendo-se das dificuldades que o operador encontra e das respectivas funcionalidades que podem acelerar o procedimento de teste, pode-se seguir para o desenvolvimento da ferramenta na segunda etapa do trabalho de graduação.

Iniciou-se o projeto da ferramenta e estão sendo definidas características como a interface, quais informações serão exibidas, e particularidades de implementação. A linguagem de programação que será utilizada para o desenvolvimento da ferramenta será o C++, no ambiente Microsoft Visual Studio 2013.

7 REFERÊNCIAS

Han, S.∗,Song, J.∗ , Zhu, X.∗ , Mok, A. K.∗ , Chen, D.† , Nixon, M.†, Pratt, W.‡, Gondhalekar, V.‡ (2009) “Wi-Wi-HTest: Compliance Test Tool for Real-Time WirelessHARTTM Mesh Network Devices”, http://apps.cs.utexas.edu/tech_reports/reports/tr/TR-2009.pdf, Junho.

HART Communication Foundation (2014a) “Wireless How it works”, http://pt.hartcomm.org/hcp/tech/wihart/wireless_how_it_works.html, Junho.

HART Communication Foundation (2014b) “Wireless Overview”, http://pt.hartcomm.org/hcp/tech/wihart/wireless_overview.html, Junho.

Kurose, J. F. e Ross, K. W. (2006) “Redes de Computadores e a Internet: Uma abordagem top-down”, Trad. 3ª edição.

Petersen, S. e Carlsen, S. (2009) “Performance Evaluation of WirelessHART for Factory Automation”, http://robotics.eecs.berkeley.edu/~pister/290Q/Papers/HART/Petersen%20WirelessHART.PDF, Junho.

Rech, J. R. e Netto, J. C. (2012) “Desenvolvimento de um gerente de rede WirelessHART”, http://www.lume.ufrgs.br, Junho.