UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Contribuições em Escalonamento e Análise de Desempenho de Redes WirelessHART

Marcelo Henrique Ramalho Nobre

Orientador: Prof. Dr. Luiz Affonso Henderson Guedes de Oliveira

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Computação da UFRN como parte dos requisitos para obtenção do título de Doutor em Ciências.

Número de Ordem do PPgEEC: D156
Natal, RN, Novembro de 2015

Seção de Informação e Referência

Catalogação da publicação na fonte. UFRN / Biblioteca Central Zila Mamede

Nobre, Marcelo Henrique Ramalho.

Contribuições em Escalonamento e Análise de Desempenho de Redes WirelessHART. – Natal, RN, 2015.

115f. ; il.

Orientador:Prof. Dr. Luiz Affonso Henderson Guedes de Oliveira.

Tese (Doutorado) – Universidade Federal do Rio Grande do Norte. Centro de Tecnologia. Programa de Pós-Graduação em Engenharia Elétrica.

1. Redes sem fio – Tese. 2. WirelessHART – Tese. 3. Escalonamento – Tese. 3. Roteamento – Tese. 4. Network simulator 3 – Tese. I. Oliveira, Luiz Affonso Henderson Guedes de. II. Universidade Federal do Rio Grande do Norte. III. Título.

RN/UF/BCZM CDU 004.725.5

Contribuições em Escalonamento e Análise de Desempenho de Redes WirelessHART

Marcelo Henrique Ramalho Nobre

Tese de Doutorado aprovada em 23 de Novembro de 2015 pela banca examinadora composta pelos seguintes membros:

Prof. Dr. Luiz Affonso H. Guedes de Oliveira (orientador) ..... DCA/UFRN

Prof. Dr. Carlos Manuel Dias Viegas ..... DCA/UFRN

Prof. Dr. Ivanovitch Medeiros Dantas da Silva ..... IMD/UFRN

Prof. Dr. Ivan Muller ... Universidade Estadual do Rio Grande do Sul/UERGS

Prof. Dr. Daniel Gouveia Costa ..... Universidade Estadual de Feira de Santana/UEFS

A minha família, todos os meus amigos e colegas, que me ajudaram ao longo de toda a jornada da engenharia da computação.

Agradecimentos

A minha família, pelo apoio durante o curso. Em especial aos meus pais e aos meus irmãos.

Ao meu orientador, Affonso.

A todo os colegas do LII e UPLAB.

A CAPES pelo apoio financeiro.

Resumo

A comunicação sem fios é uma tendência atual no ambiente industrial e nessa tendência temos o *Wireless*HART como uma das principais tecnologias. Com essa situação, é natural que melhorias no desempenho dessa tecnologia sejam buscadas e um dos principais caminhos para isso passa pelo desenvolvimento de algoritmos de escalonamento e roteamento. A primeira contribuição dessa tese é uma revisão da literatura sobre as principais soluções em escalonamento e roteamento desenvolvidas especificamente para a tecnologia *Wireless*HART. Porém, a principal contribuição dessa tese é proposição de um novo algoritmo de escalonamento de mensagens em redes *Wireless*HART, chamado Escalonamento Flow, que objetiva melhorar aspectos de flexibilidade e de utilização do superframe. Como terceira contribuição, foi desenvolvido e utilizado, com propósito de validação, um módulo de simulação para o Network Simulator 3 (NS-3), que modela aspectos como posicionamento, atenuação de sinal e consumo de energia, além de prover simulações mais precisas por meio de configurações de erro individuais para cada *link*. Este módulo também possibilita a geração do *superframe* de escalonamento a partir do grafo de roteamento utilizando os algoritmos Flow e Han. Para a validação do novo algoritmo são realizados experimentos comparativos entre o algoritmo Han e algoritmo Flow, avaliando critérios de alocação de *links*, *delay* e taxa de ocupação de *superframe*. Para validação da camada física do módulo de simulação, o escalonamento e o roteamento são configurados estaticamente e foram desenvolvidos experimentos de confiabilidade e consumo de energia com topologias validadas na literatura e com variações de probabilidades de erro.

Palavras-chave: Redes Sem fios, WirelessHART, Escalonamento, Roteamento, Network Simulator 3.

Abstract

Wireless Communication is a trend in the industrial environment nowadays and on this trend, we can highlight the WirelessHART technology. In this situation, it is natural the search for new improvements in the technology and such improvements can be related directly to the routing and scheduling algorithms. In the present thesis, we present a literature review about the main specific solutions for Routing and scheduling for WirelessHART. The thesis also proposes a new scheduling algorithm called Flow Scheduling that intends to improve superframe utilization and flexibility aspects. For validation purposes, we develop a simulation module for the Network Simulator 3 (NS-3) that models aspects like positioning, signal attenuation and energy consumption and provides an link individual error configuration. The module also allows the creation of the scheduling superframe using the Flow and Han Algorithms. In order to validate the new algorithms, we execute a series of comparative tests and evaluate the algorithms performance for link allocation, delay and superframe occupation. In order to validate the physical layer of the simulation module, we statically configure the routing and scheduling aspects and perform reliability and energy consumption tests using various literature topologies and error probabilities.

Keywords: Wireless Networks, WirelessHART, Scheduling, Routing, Network Simulator 3

Sumário

Lista de Publicações

Nobre, Marcelo, Ivanovitch Silva & Luiz Affonso Guedes (2015*b*), 'Routing and scheduling algorithms for WirelessHART networks: A survey', *Sensors* **15**(5), 9703–9740. **doi**:10.3390/s150509703

Nobre, Marcelo, Ivanovitch Silva & Luiz Affonso Guedes (2015), 'Performance evaluation of wirelesshart networks using a new network simulator 3 module', *Computers and Electrical Engineering* (41), 325–341. **doi**:10.1016/j.compeleceng.2014.05.005

Nobre, Marcelo, Ivanovitch Silva & Luiz Affonso Guedes (2014), 'Reliability evaluation of wirelesshart under faulty link scenarios'. *12th IEEE International Conference on Industrial Informatics* 2014, pp. 676–682. **doi**:10.1109/INDIN.2014.6945595

Nobre, Marcelo, Ivanovitch Silva & Luiz Affonso Guedes (2014), Avaliação de Confiabilidade e Consumo de Energia em Redes WirelessHART. *XX Congresso Brasileiro de Automática* 2014.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

ACK Acknowledge

ASN Absolute Slot Number

BER Bit Error Rate

C-LLF Conflict-aware Least Laxity First

CCA Clear Channel Assessment

CLP Controlador Lógico Programável

COOJA Contiki OS Java

CRC Cyclic Redundancy Check

CSMA Carrier Sense Multiple Access

CWSA Contention Window Size Adjustment

D-MSR Distributed Network Management Scheme for Real-Time applications

DCS WirelessHART-based Distributed Control Systems

DLPDU Data-Link Protocol Data Unit

DSSS Direct Sequence Spread Spectrum

EIRP Equivalent isotropically radiated power

ERP Enterprise Resource Planning

FEC Forward Error Correcting

FHSS Frequency Hop Spread Spectrum

IEC International Electrotechnical Commission

ISM Industrial, Scientific and Medical

LLC Logical Link Control

LLT Least Laxity First

MAC Medium Access Control

MES Manufacturing Execution Systems

MIC Message Integrity Code

MMD Multi-service Data Denoising

MMNS MRF-based Multi-service Node Scheduling

MRF Markov Random Field

NAM Network Animator

NIC Network Interface Cards

NS-2 Network Simulator 2

NS-3 Network Simulator 3

O-QPSK Offset Quadrature Phase-Shift Keying

OTCL Object Oriented Tool Command Language

OUI Organizationally Unique Identifier

PAN Personal Area Networks

PDU Process Data Unit

PER Packet Error Rate

PIB PAN Information Base

PR Publish Rate

RISSF Rede Industrial de Sensores Sem Fio

RSD Representative node Selection and service Determination

SFD Start of Frame Delimiter

SOM Start of the Message

TAMS Traffic-aware message scheduling

TclCl Tcl with Classes

TCP Transmission Control Protocol

TDMA Time Division Multiple Access

TPDU Transport Protocol Data Unit

Capítulo 1

Introdução

Redes de sensores sem fios são uma tendência em comunicação de dados, sendo aplicadas nas mais diversas áreas, incluindo biomedicina, automação residencial, monitoramento ambiental e monitoração por vídeo [Costa & Guedes 2010]. Consequentemente, os ambientes industriais também estão se beneficiando de tal tecnologia, visando os benefícios providos pela mesma como, por exemplo, economia de cabos e flexibilidade. O foco principal desta tese é o estudo de desempenho e confiabilidade de Redes Industriais de Sensores Sem Fios. Mais especificamente, o estudo de protocolos de escalonamento de mensagens em redes tipos *Wireless*HART.

A rede de comunicação sem fios aplicada ao ambiente industrial é denominada nesta tese de Rede Indutrial de Sensores Sem Fios (RISSF), sendo esta formada por três elementos básicos:

- Sensores: Dispositivos que medem as variáveis envolvidas no processo, tais como: temperatura, nível, pressão etc.
- Atuadores: Dispositivos que realizam ações para modificar as variáveis do processo e se obter o comportamento desejado.
- Controladores: Elementos que, a partir dos dados obtidos pelos sensores, da estratégia de controle que se deseja aplicar e dos comandos dos operadores do sistema, determinam dinamicamente as ações dos elementos do sistema para que o objetivo do processo de produção seja atingido.

Atualmente há, basicamente, dois padrões para o RISSF: O *Wireless*HART [(IEC) 2010] e o ISA100 [(IEC) 2012]. Ambas utilizam o padrão IEEE802.15.4 como camada física. O *Wireless*HART é uma evolução da tecnologia de HART (que é tradicionalmente cabeada) publicada na especificação HART 7.1 e teve seu padrão aprovado pela IEC no início de 2010. O ISA100 Foi publicada em 2009 pela *International Society of Automation*

(ISA) e foi projetada para com base em necessidade de escalonamento dos usuários e não em uma tecnologia preexistente como feito com o *Wireless*HART.

Com o objetivo de descrever os principais desafios das redes industriais, este capítulo introdutório aborda um breve histórico das RISSF, a importância da utilização deste tipo de rede e os desafios enfrentados no atual contexto dessa tecnologia. Por fim, serão apresentados os objetivos, a motivação e as contribuições da presente tese, bem como este documento será organizado nos capítulos subsequentes

1.1 Histórico das Redes Industriais sem fios

As redes de chão-de-fábrica são as que operam no nível das plantas industriais, fornecendo a comunicação entre os dispositivos mais básicos (sensores, atuadores) e os dispositivos de alto nível (controladores lógico programáveis e computadores de supervisão). Quando comparadas com as redes de computadores tradicionais, as redes de chão-de-fábrica apresentam diferenças principalmente associadas aos dados transmitidos, padrão de tráfego, confiabilidade das aplicações e exigências rígidas de tempo real (tempo real firme) [Gungor & Lambert 2006]. Em um ambiente tipicamente industrial, as informações a serem monitoradas estão relacionadas com o estado dos equipamentos (ligado, desligado, ocioso, bloqueado), alarmes e variáveis de processos. Em geral, essas informações são transmitidas por pacotes de dados cujo tamanho é inferior a 100 bytes, enquanto que as taxas de transmissão são limitadas a algumas centenas de kilobits por segundo [Silva et al. 2010*a*]. Por outro lado, em uma rede local de propósito geral, os pacotes de dados são superiores a 100 bytes e as taxas de transmissão podem chegar a 10 Gbps [Lin et al. 2009]. Em relação aos intervalos de transmissão em uma aplicação industrial é comum os dispositivos serem configurados com intervalos de 1 a 4 segundos (aplicações de controle de processos) a alguns minutos (aplicações de monitoramento). Em geral, os intervalos de transmissão têm taxa constante com exceção dos alarmes, que dependem de variáveis estocásticas. Finalmente, a confiabilidade fim-a-fim é um dos principais itens a ser considerado pelas aplicações industriais. De modo geral, a comunicação entre os dispositivos de chão-de-fábrica e as aplicações de gerenciamento deve atender requisitos rígidos de confiabilidade e determinismo [Sauter et al. 2011]. Em uma rede de computadores tradicional, requisitos de confiabilidade e tempo real são mais relaxados quando comparado com as exigências das aplicações industriais. Outra característica antagônica está relacionada com o período no qual as instalações dessas redes são válidas. Em uma rede de chão-de-fábrica é esperado que sua estrutura seja utilizada por um período de pelo menos até 10 anos, enquanto que em uma rede tradicional esse período é de aproximada-

mente 3 anos [Sauter 2010].

A diferença entre as redes de chão-de-fábrica e as redes tradicionais são dadas principalmente pela limitação das tecnologias utilizadas e dos requisitos das aplicações. Entretanto, o avanço tecnológico tem criado uma estrutura adequada para estender as redes originalmente projetadas para serem instaladas em escritórios (Ethernet e soluções sem fio) de ambientes industriais. Para compreender melhor esse contexto é necessário observar toda a evolução das redes de chão-de-fábrica, desde as tecnologias mais primitivas até a utilização das redes sem fio. A Figura 1.1 é um exemplo de tal abordagem, em que as diferentes tecnologias de comunicação para redes de chão-de-fábrica são organizadas cronologicamente de acordo com essa evolução.

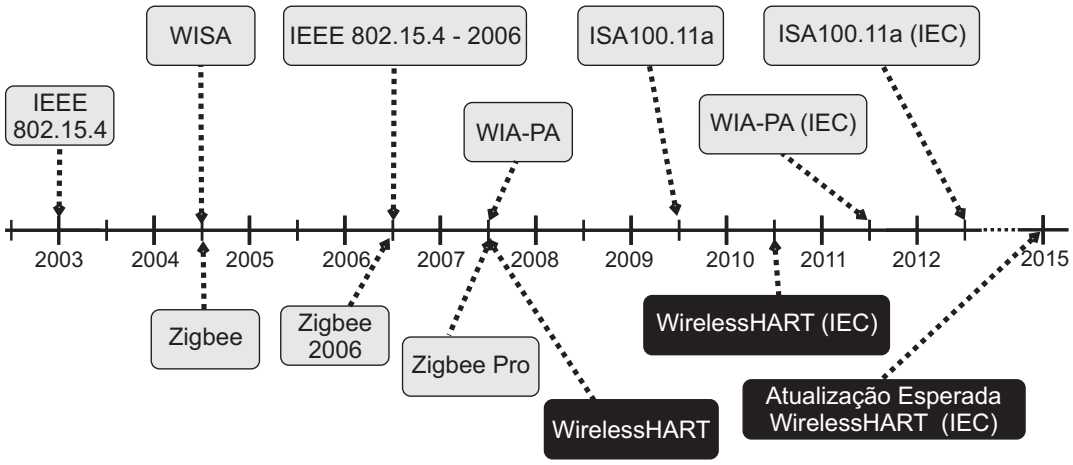


Figura 1.1: Evolução das redes de Chão-de-Fábrica.

1.1.1 Primeiras Tecnologias

Embora a origem das redes de chão-de-fábrica datam de 30 anos atrás, os conceitos por trás dessas redes são muito mais antigos e remontam às redes Telex e às primeiras tecnologias usadas nas redes de telecomunicações (V.21 e X.21) [Sauter 2005]. A evolução dos microprocessadores permitiram que esses sistemas migrassem de uma arquitetura analógica para uma arquitetura digital, tornando-se possível o desenvolvimento de tecnologias com maiores taxas de transmissão. As redes de chão-de-fábrica foram criadas com a finalidade de disponibilizar as informações presentes no mais baixo nível das plantas industriais para toda a companhia. Essa demanda resultou em um modelo hierárquico representado pela pirâmide da automação, como descrito na Figura 1.2. Nesse modelo, as camadas inferiores são interligadas pelas redes de chão-de-fábrica, enquanto que as camadas superiores estão sob influência das redes corporativas. A comunicação entre as

redes de chão-de-fábrica e as redes corporativas são geralmente realizadas nas camadas intermediárias através de dispositivos específicos para essa finalidade.

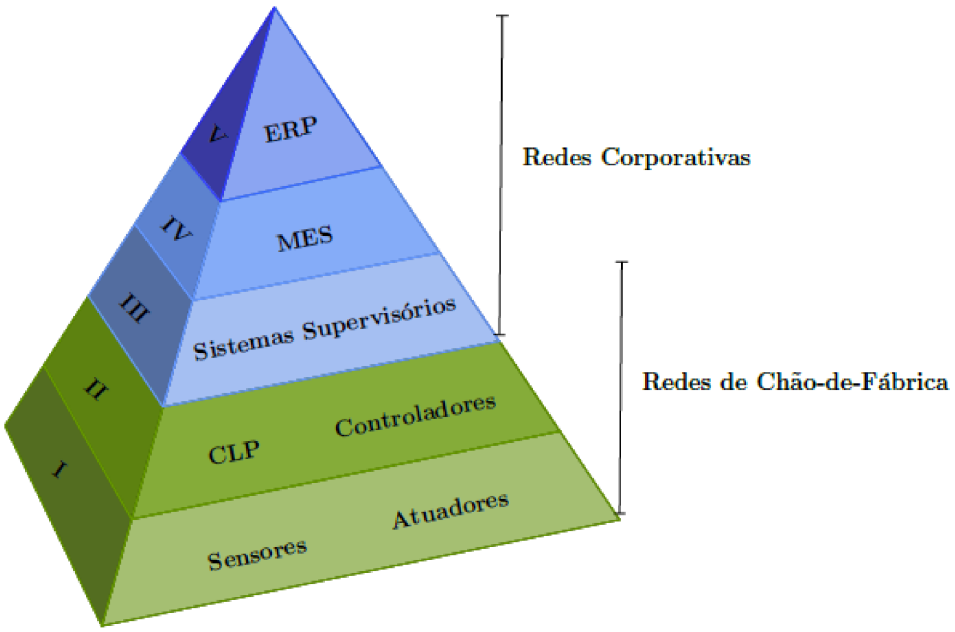


Figura 1.2: Pirâmide da automação - modelo hierárquico representando os diversos níveis da automação industrial.

Durante as décadas de 80 e 90, diversas soluções proprietárias para redes de chão-de-fábrica foram desenvolvidas. Essas soluções eram dedicadas a aplicações específicas e redes de diferentes fabricantes eram incompatíveis. A maioria dessas soluções proprietárias desaparecem devido à inviabilidade econômica de manter um protocolo muito especializado para um nicho específico de aplicações [Thomesse 2005]. Motivado por tal contexto, surgiu a necessidade de uma padronização internacional das redes de chão-de-fábrica. Se analisarmos a evolução dessas redes do ponto de vista da padronização, é possível perceber a influência direta do modelo OSI/ISO (Interconexão de Sistemas Abertos/Organização Internacional para Padronização). Esse modelo serviu de base para que protocolos fossem desenvolvidos com a finalidade de comunicar redes de diferentes fabricantes. Um dos primeiros protocolos desenvolvidos foram o MAP e Mini-MAP. Entretanto, a complexidade de implementação inviabilizou a sua utilização em larga escala [Schutz 1988].

No geral, a tentativa de criar um protocolo universal para as redes de chão-de-fábrica fracassou. Essa busca tecnológica durou aproximadamente 16 anos (1986-2002) [Felser & Sauter 2002]. As tentativas para padronizar uma solução universal tornaram-se uma

questão política, em que cada país desejava garantir sua solução como a universal, do que uma discussão técnica. Inicialmente, França (FIP) e Alemanha (PROFIBUS) disputaram a escolha da solução universal para as redes de chão-de-fábrica. Obviamente, cada país defendia sua própria solução. Entretanto, o FIP e o PROFIBUS propuseram padrões que se complementavam. O primeiro utilizava uma solução centralizada, enquanto que o segundo propôs uma solução distribuída. Algumas tentativas foram feitas com o objetivo de incrementar o PROFIBUS com as funcionalidades propostas pelo FIP e vice-versa (ISP, WorldFIP). Em 1995, diversas empresas, na sua grande maioria empresas americanas, decidiram não esperar por uma solução universal oriunda da combinação entre o FIP e PROFIBUS, criando a definição de um novo protocolo, o Foundation Fieldbus. Naquele momento, os países europeus que detinham tecnologias nacionais para redes de chão-de-fábrica estavam receosos de que suas soluções não pudessem se tornar padrões universais. Do ponto de vista econômico não era viável desenvolver um novo protocolo "do zero" que não fosse compatível com as respectivas soluções nacionais. Dessa forma, o Comitê Europeu de Normalização Eletrotécnica (CENELEC) compilou todos os padrões nacionais europeus para um único documento, onde cada padrão é visualizado com uma parte isolada ou um sistema independente. Em paralelo, a Comissão Eletrotécnica Internacional (IEC), através da Foundation Fieldbus, criou sua própria especificação. A coexistência do padrão europeu com a especificação do IEC criou um cenário conhecido como a "guerra das redes de chão-de-fábrica". No final, o objetivo de criar um protocolo universal para as redes de chão-de-fábrica foi abandonado com o surgimento dos padrões IEC 61158 e IEC 61784, os quais acomodavam todos protocolos de redes de chão-de-fábrica existentes [Felser 2002]. É importante lembrar que nessa época todos os referidos protocolos usavam cabos como meio de transmissão.

1.1.2 Eliminação do cabeamento e vantagens das RISSF

O surgimento das tecnologias de redes industriais sem fio foi uma evolução natural das primeiras tecnologias de comunicação industrial desenvolvidas, que usavam cabos como meio de transmissão. A proposta de eliminar o cabeamento e utilizar um novo paradigma na transmissão dos dados em ambientes industriais não é recente, por exemplo, [Lessard & Gerla 1988] desenvolveram um dos primeiros trabalhos nessa área na tentativa de comunicar dispositivos industriais por infravermelho. Com a evolução das tecnologias de comunicação, novos mecanismos foram desenvolvidos para garantir a confiabilidade das redes sem fio (modulação e codificação, escalonamento determinístico, saltos de canais e topologias redundantes), tornando sua utilização acessível aos ambientes industriais

[Gungor & Hancke 2009] [Han et al. 2011*a*].

Antes domínio dos sistemas de comunicação cabeados como os protocolos *foundation fieldbus* e o HART, o ambiente industrial vem sendo modificado à medida que as tecnologias sem fios progridem no sentido de prover as mesmas vantagens das sistemas cabeados, com a adição das suas vantagens naturais. Segundo [Akerberg, Gidlund, Lennvall, Neander & Bjorkman 2011], as maiores vantagens das redes sem fios aplicadas na indústria seriam o custo, a flexibilidade, as novas aplicações e a disponibilidade.

A minimização dos custos de produção é uma constante dentre os investidores, e o incentivo para a adoção de RISSF nos processos de automação é a facilidade e o baixo custo de instalação em relação à contrapartida cabeada. Segundo [Akerberg, Gidlund, Lennvall, Neander & Bjorkman 2011] o custo de instalação de cabos em uma nova planta industrial, esta na faixa de 200 dólares por metro e salta para 1000 dólares por metro em instalações *offshore*. Além disso, cabos envelhecem, falham ou quebram, necessitando de serviços de inspeção, testes e diagnósticos de problemas, além de custos, mão de obra e materiais para eventuais melhorias na infraestrutura. Segundo [Colpo & Mols 2011], o uso de equipamentos sem fio pode reduzir os custos de instalação entre 50-90%, quando comparados com cenários em que dispositivos cabeados são utilizados.

Quanto à flexibilidade, muitas variáveis de processo secundárias são dificilmente medidas, e equipamentos giratórios críticos ainda são não instrumentados. Com a utilização das RISSF, foi possível se obter informações que antes era economicamente inviável, de tal modo que estes novos dados podem ser utilizados para melhorar o processo com relação à quantidade e à qualidade, com a finalidade de reduzir a chance de falhas mecânicas. Além disso, as RISSF permitem a medição de variáveis temporariamente sem a necessidade da instalação de cabos adicionais.

Podemos destacar também as novas aplicações possibilitadas pelo uso das RISSFs como por exemplo robôs atuadores móveis, localização de recursos, segurança, e interligação de sistemas de controle remotos ou isolados, controle sem fio da planta e conectividade com equipamentos selados (*"sealed-for-life"equipments*).

Aplicações industriais necessitam de disponibilidade e determinismo para evitar consequências graves como danos a integridade física de funcionários, explosões e perdas materiais. As RISSF oferecem redundância e capacidade de manutenção preventiva e recuperação de falhas. Isso pode ser atingido por utilizar redes em malha, possibilitando a criação de rotas redundantes e sistemas de controle.

1.2 Desafios atuais para as RISSF

Apesar das vantagens mostradas anteriormente, [Akerberg, Gidlund, Lennvall, Neander & Bjorkman 2011] afirma que alguns requisitos importantes não podem ser encontrados de maneira eficiente na implementações atuais ou até mesmo não existam nas RISSF de larga escala implantadas. Para que as RISSF sejam adotadas, elas devem atender todos os requisitos atendidos pela sua contrapartida cabeada, uma vez que os investidores não têm interesse em manter duas estruturas paralelas com a mesma função. Os desafios que podemos ressaltar são proteção de pessoas e estrutura, segurança de dados, disponibilidade, latência, falta de suporte a atuadores, integração ao sistema já existente, tamanho da rede, coexistência com outras tecnologias, colisões de transmissão e consumo de energia.

1.2.1 Segurança física

A segurança de pessoas, do ambiente e da propriedade deve ser sempre uma prioridade. Nos processos de automação, algumas funções são de segurança crítica por natureza, mas não são a maioria. Isto não significa que apenas as funções de segurança críticas sejam cuidadosamente projetadas desenvolvidas e validadas, uma vez que o funcionamento da planta depende do funcionamento adequado (dentro das especificações) de todo o sistema. Mesmo uma funcionalidade não sendo crítica em relação à segurança, podem haver perdas consideráveis caso os equipamentos não sejam projetados para reduzir o risco de situações perigosas ou fora de controle. A prevenção de situações descontroladas é extremamente importante, por exemplo, se o valor de referência para uma válvula não pode ser transmitido, tal válvula deve retornar a um estado seguro depois de um certo tempo de espera. Este tempo é determinado pelo quanto este processo pode tolerar o mau funcionamento do atuador antes de se entrar em um estado perigoso, podendo variar de milissegundos a segundos. Além disso o sistema de controle deve detectar falha na comunicação do sistema e sinalizar tal falha, de modo que os outros equipamentos dependentes deste processo evitam situações de perigo provocadas pela propagação do erro.

Um dos piores cenários que podem ocorrer é o de inconsistência entre a situação real da planta e o que é percebido pelo sistema de controle e pelo operador. Isto implica que ciclos de trabalho muito pequenos, com o intuito de poupar energia, podem fazer com que o sistema de controle e os operadores não tenham sinais de *keep alive* suficientes do dispositivos ou manter a taxa de atualização necessária. Um trabalho em comunicação crítica de segurança com *Wireless*HART foi realizado em [Akerberg, Reichenbach & Bjorkman 2010] e o resultado principal é o de que RISSF s devem ter comunicação

sincronizada e determinística no *uplink* e *downlink* para se evitar falsos positivos quanto aos *timeouts* de falha.

1.2.2 Segurança de Dados

Como o sinal de comunicação é transmitido pelo ar, qualquer dispositivo de rede que esteja no alcance do rádio conseguirá captar o sinal. Assim, torna-se fundamental a utilização de um nível elevado de segurança na tentativa de evitar o acesso indevido a informações sigilosas. No extremo, pessoas não autorizadas podem aproveitar da ausência de segurança para injetar pacotes nocivos à rede com o intuito de realizar algum ataque ou roubar informações [Kai-Di Chang 2012].

A maioria das informações transmitidas por um dispositivo de campo é normalmente um valor normalizado do processo medido, que vai de 0 a 100% da escala do instrumento de medição e, em alguns casos, com a unidade de medida. O sistema de controle coleta sinais do estado do processo e, com base nesses sinais e na estratégia de controle, transmite os valores de referência para os atuadores. Entretanto, o ponto principal é que esta informação transmitida não é confidencial. As informações confidenciais são as estratégias de controle contidas no sistema de controle e não são transmitidas aos dispositivos de campo. Contudo, do ponto de vista da segurança de informação, autenticidade, integridade, disponibilidade e irretratabilidade (não repudiação) são objetivos importantes da segurança. Segundo [Akerberg, Gidlund, Lennvall, Neander & Bjorkman 2011], na situação atual das RISSF a confidencialidade, autenticação e integridade são fornecidos. Mas otimizações podem ser feitas com respeito à segurança para a economia de energia, latência e *overhead* de segurança.

Outro desafio que podemos ressaltar é o de como integrar os mecanismos de segurança no sistema de automação como um todo, realizando um gerenciamento de chaves de segurança e controlando de maneira segura a reposição de dispositivos na rede.

1.2.3 Disponibilidade

O uso de tecnologias de comunicação sem fio em ambientes industriais sempre foi visto com grande ceticismo por parte das companhias. Esse cenário foi criado principalmente pela baixa confiabilidade do canal de comunicação, haja vista que os equipamentos são instalados em áreas sujeitas à influência de agentes externos (ruídos, interferências, clima adverso, obstáculos naturais), que podem provocar erros em taxas superiores aos das tecnologias cabeadas [Bai & Atiquzzaman 2003]. Outros erros no canal de comunicação são decorrentes da atenuação do sinal (motivada principalmente pela perda de

potência devido à distância entre o emissor e receptor) e do fenômeno de múltiplos caminhos (devido à reflexão, difração e dispersão do sinal transmitido, múltiplas cópias do dado podem sofrer interferência construtiva ou destrutiva na recepção).

Em geral, erros em comunicações sem fio são transientes, ou seja, o canal de comunicação fica ruim por um tempo e depois retorna à normalidade. Por outro lado, em tecnologias cabeadas o erro mais frequente é o permanente, devido a falhas nos cabos, conectores ou outros componentes. Um segundo ponto a ser levado em consideração na inserção de tecnologias de comunicação sem fio em ambientes industriais está relacionado à natureza do meio de propagação.

Tendo essa situação, a disponibilidade da rede é de grande importância na produção industrial em grande escala. Mesmo erros de transmissão passageiros podem causar interrupções na produção. Isto se deve principalmente porque o processo precisaria ser parado de maneira controlada no caso de um problema de comunicação e necessitaria de várias horas para atingir novamente o nível de produção máximo. Redes auto estruturantes (*self-healing*) são atraentes para a automação industrial por diversos motivos como, por exemplo, redundância e disponibilidade.

Entretanto, é comum na literatura que o protocolo de roteamento das redes industriais lide com topologias em malha contendo milhares ou dezenas de milhares de nós. Além disso, também é assumido que os dispositivos são alimentados por baterias, o que torna relevante protocolos de roteamento que levem em consideração o nível de energia das estações para o balanceamento da carga de dados sobre os nós de maneira adequada. Segundo [Akerberg, Gidlund, Lennvall, Neander & Bjorkman 2011], a primeira afirmação não é correta, uma vez que, apesar de que haja milhares de dispositivos para se comunicar em uma planta, eles não pertencem a mesma rede com respeito à disponibilidade. Os nós são distribuídos por um conjunto de controladores de processos, divididos em diversas seções de processos, de modo a não parar completamente a produção em caso de falhas. Além disso, protocolos de roteamento otimizados para poupar energia podem ter severos impactos negativos na latência e na performance da comunicação em tempo real em redes em malha no caso de *links* com desvanecimento. Na automação industrial, as redes em malha podem se beneficiar das alterações rápidas no roteamento, no caso de desvanecimento, por exemplo, enquanto ainda atendam aos requisitos de tempo real.

1.2.4 Latência/Retransmissão

Devido à natureza da automação industrial, os dados transmitidos em redes de campo são válidos por um curto período de tempo. Caso o dado seja entregue com um atraso

relevante, o este dado pode perder sua utilidade para a maioria dos sistemas em tempo real. Portanto, novos dados produzidos devem ser propagados pela rede, ao invés de se garantir a entrega de todos os pacotes mais antigos. Esta é uma importante área de pesquisa, especialmente em RISSF, onde podem existir redes em malha, *multi-hop* e comunicação síncrona bidirecional entre nós. Além disso, sistemas de automação devem instalar dados de configuração em dispositivos de campo tanto na fase de inicialização da rede quanto durante a operação, sendo feito de maneira fim-a-fim com suporte a retransmissão e confirmação para se evitar perda de informações. Para se diminuir o número de retransmissões, nas RISSF é possível utilizar técnicas de controle de erros como a correção antecipada (*Forward Error Correcting*-FEC), que codifica de maneira redundante os dados utilizando um código corretor de erro [Li et al. 2010]. Nas redes baseadas no padrão IEEE 802.15.4, o FEC foi omitido devido ao alto consumo de energia na operação de decodificação. No entanto, usando o FEC, o consumo global de energia vai se tornar menor, uma vez que será gasto menos energia com retransmissões e reescalonamento [Vuran & Akyildiz 2009].

Com relação ao aspecto das retransmissões em redes em malha e situações de *multihop*, é necessário que os dados sejam entregues na ordem correta. Podem ocorrer casos em que dados não sejam recebidos na ordem correta caso pacotes em atraso não sejam descartados na malha antes da chegada do novo dado periódico. As consequências podem ser consideráveis, uma vez que estas informações são utilizadas, por exemplo, para iniciar/parar um motor elétrico.

1.2.5 Falta de suporte a atuadores

Uma grande vantagem das RISSF é a de que não é necessário o cabeamento de cada sensor e atuador ao controlador. Normalmente, diferentes seções do processo são distribuídas por vários, e por vezes redundantes, controladores para a evitar que uma falha em uma parte do processo afete outras seções. A razão principal é a de limitar as consequências em caso de falha, e aumentar a disponibilidade da planta através de *buffers* de dados entre as diferentes seções do processo. Contudo, os atuadores ganhariam a mesma vantagem dos controladores em relação a este caso. Hoje em dia, a maioria dos padrões não dá suporte a atuadores (permitem apenas leitura de sensores) e, portanto, limita a suas vantagens, uma vez que uma estrutura paralela para atuadores deve ser projetada, instalada e mantida. Isso, por fim, também reduz os benefícios da RISSF, uma vez que o são necessários custos adicionais com infraestrutura de cabos. Suporte a atuadores necessita de comunicação bidirecional determinística, bem como estados seguros em caso de falhas.

Um dos maiores argumentos contra o suporte a atuadores atualmente é o de que atuadores necessitam estar ligados à rede elétrica, uma vez que uma bateria não é suficiente para suprir energia para o atuador.

1.2.6 Integração com sistemas existentes

Para se manter uma integração eficiente das RISSF em infraestruturas de automação de plantas já existentes devemos considerar como ponto crítico os *gateways*. Atualmente, existe um pequeno número de fornecedores de *gateways* e estes propõem soluções proprietárias que impedem uma integração mais eficiente e aberta com infraestruturas legadas. Todas as soluções disponíveis comercialmente fornecem configuração via Internet, onde tudo é realizado manualmente, onerando a engenharia, auditorias e a manutenção do sistema. Hoje em dia a maioria dos sistemas cabeados tem serviço de *download* de configurações para simplificar o esforço de integração. O que atualmente está faltando é uma abordagem padronizada para a integração da RISSF aos diferentes padrões de cabeamento [Akerberg, Gidlund, Lennvall, Neander & Bjorkman 2011]. O esforço de padronização pode ser exemplificados pela proposta de integração [Akerberg, Gidlund, Lennvall, Neander & Björkman 2010]. Portanto, é importante que os serviços providos pela RISSF sejam possíveis de ser integrados eficientemente ao sistema de automação para uma transição suave entre as tecnologias cabeada e sem fios, se mantendo a simplicidade de instalação, auditoria e manutenção.

1.2.7 Escalabilidade da rede

Outro importante requisito para as RISSF é a escalabilidade e a capacidade da rede de se adaptar às mudanças no tamanho da rede. Sem este tipo de suporte, a performance da rede vai se degradar significativamente com o crescimento do número de dispositivos ligados. Na maioria das linhas de pesquisa é previsto que o gateway seja escalável para suportar até 1000 nós sensores, contudo esse número pode variar com as técnicas utilizadas (por exemplo, se a rede é TDMA ou utiliza múltiplos caminhos). Esta afirmação não é válida em um contexto de automação industrial segundo [Akerberg, Gidlund, Lennvall, Neander & Bjorkman 2011], pois neste tipo de ambiente uma rede irá conter no máximo 50 nós para que sejam atingidos as taxas de atualização requeridas.

1.2.8 Coexistência e evitar interferências

Plantas industriais geralmente contém várias tecnologias de comunicação *wireless* que operam na mesma frequência, a maioria na banda ISM de 2,4 GHz. Logo é importante que uma solução possa coexistir nesse ambiente de rádio com interferências e que também minimize a sua própria interferência nas outras tecnologias. A primeira medida contra a interferência é a adoção de diferentes esquemas de acesso ao canal (tempo, espaço e frequência), e também de algumas novas técnicas como o cancelamento de interferência, gerenciamento efetivo de recursos de rádio e rádios configuráveis por software. Tecnologias como o *Wireless*HART utilizam a divisão por tempo (TDMA) em conjunto com saltos em canais e algoritmos de escalonamento. Segundo [Akerberg, Gidlund, Lennvall, Neander & Bjorkman 2011], é importante garantir os serviços de tempo real das RISSF e portanto a pesquisa em técnicas de escalonamento em tempo real são de grande importância.

1.2.9 Consumo de energia

Em pesquisas, o consumo de energia e a captação de energia são tópicos emergentes atualmente. Entretanto, devido às taxas de atualização requeridas nos processos de automação, é difícil se realizar economia de energia por redução do ciclo de trabalho nos dispositivos de campo sem fios. Além disso, ainda existe a presença dos atuadores, muitas vezes pneumáticos, que respondem por parte considerável do consumo energético. Portanto, redes completamente sem fios não estão previstas para um futuro próximo em processos de automação. Entretanto, instalações temporárias de sensores utilizadas para validação de possíveis otimizações no processos podem se beneficiar uma vez que eles podem ser sem fios de fato por operar apenas com baterias. Pesquisas com coleta de energia do ambiente vão melhorar este quadro e mais dispositivos de campo podem se tornar sem baterias de fato. Entretanto, muitos dispositivos são improváveis de se tornar sem fios com a tecnologia atual, tanto sensores (acústicos ou laser, por exemplo) e atuadores (válvulas de grande porte e controles pneumáticos). Não obstante, as vantagens ainda são significativas mesmo que as fontes de energia tenham de ser cabeadas para atuadores e sensores, pois as redes de dados não necessitam ser ligadas a todos os dispositivos por cabos em seções específicas do processo. As fontes de energias geralmente são próximas aos dispositivos de campo e provavelmente continuarão próximas, mesmo em campos abertos de produção.

Em [Akerberg, Gidlund, Lennvall, Neander & Bjorkman 2011], é proposto ainda que se imaginarmos uma situação em que todos os dispositivos de uma planta fossem man-

tidos apenas por bateria, teríamos um problema relevante que seria a manutenção das, geralmente, milhares de baterias em um processo industrial típico. Este caso seria um problema grave para indústrias das quais se espera que operem sem falhas 24 horas por dia e com poucas manutenções anuais. Além disso, os usuários finais precisariam manter estoques de baterias de vários tamanhos e capacidades para a pronta reposição de baterias esgotadas, significando que dispositivos totalmente sem fios seriam mais custosos que dispositivos que apenas se comunicam sem fios.

1.3 Motivação

Apresentada a tendência da adoção da tecnologia de comunicação sem fios na indústria e os desafios, várias propostas de soluções para as RISSF existem e outras ainda estão a ser vistas. Existem até exemplos onde redes sem fios são utilizadas como complemento para redes cabeadas tradicionais, como mostrado por [Boyes 2011]. Isso estabelece um cenário no qual dificilmente uma única solução será considerada universal para toda a indústria. É mais prudente pensarmos que várias soluções coexistirão, cada uma com seu nicho de aplicações, vantagens e desvantagens.

A literatura apresenta várias comparações entre as tecnologias sem fios. A comparação entre o *Wireless*HART e o ISA100.11a pode ser vista em [Petersen & Carlsen 2011, Wang 2011, Nixon 2012]. O padrão chinês WIA-PA é adicionado à comparação com o ISA100.11a e o *Wireless*HART em [Liang et al. 2011]. Em [Manges et al. 2012] uma perspectiva de usuário do uso da tecnologia sem fios em uma usina de energia é discutida. Além disso, [Zhu et al. 2012] apresenta um teste onde o *Wireless*HART obtém sucesso ao substituir uma rede Profibus em um *loop* fechado de controle. Dentre as tecnologias mencionadas anteriormente, o *Wireless*HART é apontada por [Song et al. 2008] como uma solução factível para redes indutriais sem fios, e será a tecnologia escolhida para ser trabalhada nessa tese.

RISSF são topologias comumente planejadas, têm baixa mobilidade de nós e alguns dispositivos da rede podem ser mantidos por baterias. Entretanto, diferentes aplicações e situações podem requerer a otimização de características específicas, tais como atraso de transmissão [Noh et al. 2008, Krishnamachari et al. 2002], consumo de energia [Silva et al. 2008] e confiabilidade dos dados [Silva et al. 2012*b*, Silva et al. 2012*a*, Costa et al. 2014, Macedo et al. 2014]. Os algoritmos de escalonamento e roteamento utilizados pela rede são importantes: eles determinam rotas, canais e a política de redundância utilizados pela rede, assim afetando diretamente as características de rede supracitadas. Além disso, existe a necessidade de se adequar a tecnologia utilizada (no caso dessa tese

o *Wireless*HART) com algoritmos de roteamento e escalonamento de modo a se otimizar as características necessárias para cada aplicação.

Tendo em vista essa situação, na presente tese iremos focar na proposição de um algoritmo de escalonamento de mensagens para redes *Wireless*HART (logo, que esteja de acordo com os requisitos da norma *Wireless*HART). Embora o escalonamento de mensagens seja uma atividade central em redes *Wireless*HART, há na literatura poucas propostas de algoritmos de fato e de código aberto em conformidade com as restrições definidas pela norma *Wireless*HART, como apontado em [Nobre et al. 2015*b*].

Tendo em vista a motivação apresentada, a nossa hipótese é a seguinte: é possível melhorar o escalonamento *Wireless*HART?

1.4 Objetivos e Contribuições da Tese

1.4.1 Objetivos Gerais

O primeiro objetivo desta tese é apresentar uma revisão bibliográfica sobre os algoritmos de escalonamento e roteamento desenvolvidos especificamente para a tecnologia *Wireless*HART. Com isso podemos lançar as bases para o desenvolvimento dos próximos objetivos.

Um segundo objetivo deste trabalho é a proposição e validação de um algoritmo de escalonamento de mensagens em conformidade com o padrão *Wireless*HART denominado de Flow, atendendo os requisitos propostos pela norma [(IEC) 2010]. O algoritmo proposto tem por objetivo acomodar todas as transmissões fim-a-fim entre a origem e destino dos fluxos de pacotes, dar suporte a múltiplas taxas de atualização dos sensores, atender os *deadlines* propostos para cada fluxo e prover uma melhor flexibilidade por meio da adoção de múltiplos critérios de escalonamento.

O terceiro objetivo é o desenvolvimento de um módulo de simulação *Wireless*HART para o NS-3 que contemple as características da tecnologia, apresente um modelo de simulação mais realista e seja de fácil uso e reprodução. Com isso, será possível auxiliar no desenvolvimento de novas técnicas que venham a melhorar o desempenho da tecnologia.

1.5 Contribuições

As contribuições desta tese para a comunidade são enumeradas a seguir.

- Estudo sistematizado sobre os principais algoritmos de roteamento e escalonamento

disponíveis para a tecnologia *Wireless*HART, incluindo um comparativo entre as principais soluções encontradas;
- Criação de um modelo de simulação da tecnologia *Wireless*HART utilizando o NS-3;
- Propor um algoritmo de escalonamento de mensagens em conformidade com a norma atual do *Wireless*HART e que melhore os aspectos de eficiência em alocação de *links* e de utilização do *superframe* das propostas atuais;
- Atestar o funcionamento das propostas de módulo de simulação e de algoritmo de roteamento através de uma análise de desempenho de redes *Wireless*HART.

1.6 Estrutura da Tese

Neste primeiro capítulo, é apresentado o contexto, motivações e contribuições da presente tese. Além deste, o documento é completado pelos seguintes capítulos.

- Capítulo 2 apresenta a tecnologia *Wireless*HART e as suas especificações mais relevantes para a tese. Dentre estas informações estão destacadas definições de roteamento e escalonamento para a tecnologia.
- Capítulo 3 faz um levantamento dos principais algoritmos de roteamento e escalonamento de mensagens desenvolvidos especificamente para a tecnologia *Wireless*HART levando em consideração características de projeto como objetivos e métricas utilizadas. Além disso, um compilado dos principais módulos de simulação do *Wireless*HART presentes na literatura é apresentado.
- Capítulo 4 detalha a nova proposta de algoritmo denominada Escalonamento Flow, bem como os resultados dos primeiros testes comparativos com o algoritmo de escalonamento Han [Han et al. 2011*b*].
- Capítulo 5 detalha o NS-3 e o módulo de simulação desenvolvido para a avaliação da tecnologia *Wireless*HART.
- Capítulo 6 mostra os resultados dos experimentos obtidos para validação do módulo de simulação *Wireless*HART para o NS-3.
- Capítulo 7 apresenta as conclusões obtidas e indicações de trabalhos futuros a serem desenvolvidos.

Capítulo 2

Redes Industriais e *Wireless*HART

A comunicação sem fio é um dos focos de pesquisa em redes industriais atualmente. Uma variedade de protocolos e tecnologias de redes sem fio podem ser usados nesse ambiente: Bluetooth, *Wireless*HART, ZigBee, Wifi e o ISA 100.11a. Desses padrões, o ZigBee, o Bluetooth e o Wifi apresentam características que dificultam o seu uso em ambientes industriais. Segundo [Song et al. 2008], o Wifi não suporta o salto entre canais e é limitado por seu consumo energético. O Bluetooth é limitado por sua baixa escalabilidade e limitação topológica (podem ser interligados apenas oito dispositivos por rede em uma topologia estrela). Por fim, os dispositivos ZigBee dividem o mesmo canal e não existe a possibilidade do salto de canais, o que os torna mais vulneráveis a ruídos persistentes, como o ruído gerado por máquinas elétricas. Entretanto, o padrão *Wireless*HART foi desenvolvido objetivando o ambiente industrial, com foco na aplicação em medição e controle de processos. O *Wireless*HART e o ISA 100 são os padrões atuais da indústria e ambos utilizam o IEEE 802.15.4 na camada física. Um comparativo entres as principais tecnologias de transmissão sem fios pode ser visto na Figura 2.1.

O padrão HART de comunicação foi criado na década de 1980, e em sua primeira versão, o *HART Field Communications Protocol*, fornecia comunicação bidirecional utilizando-se de um sinal de 4-20mA e sem comprometimento da integridade do dado medido. Nos 30 anos de existência do protocolo, o HART passou a oferecer uma gama de funcionalidades, para comunicações com ou sem fios, que incluem transmissões de dados sem solicitação, notificação de eventos, modos de transferência de dados em bloco, segurança e diagnósticos avançados. Tais diagnósticos incluem informações sobre o dispositivo, sobre o equipamento ao qual o dispositivo está atrelado e até mesmo informações do processo monitorado.

As versões mais recentes do HART (Versão 7) apresentam, entre outras características, uma inovação no sentido de implementar redes sem fio em malha através do novo protocolo de comunicação: o *Wireless*HART. Como na sua contrapartida cabeada, o *Wi-*

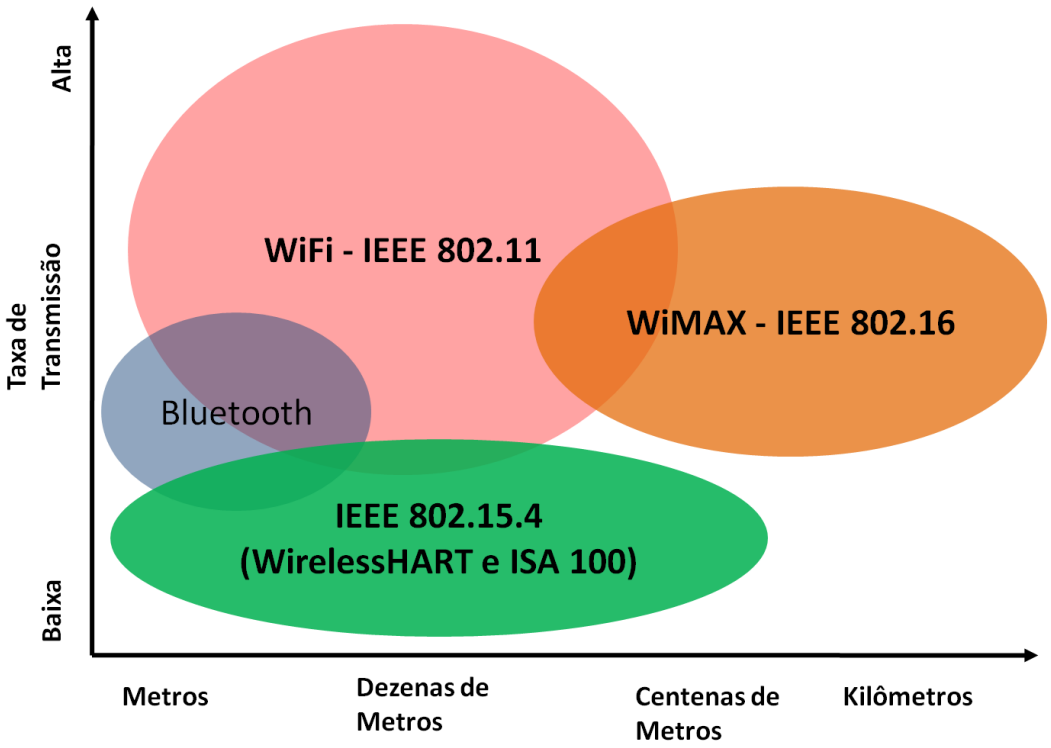


Figura 2.1: Comparativo de alcance e taxa de transmissão das principais tecnologias wireless para redes industriais.

*reless*HART tem por objetivo a comunicação com sensores e atuadores fixos, entretanto objetiva também equipamentos com sensores em partes com movimentos giratórios e sistemas flexíveis de manufatura. O *Wireless*HART aceita sistemas legados (aplicações e equipamentos HART) e reduz os custos de instalação por ser sem fios.

O *Wireless*HART se utiliza de tecnologias existentes como o próprio padrão HART, o padrão IEEE-802.15.4 [*IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs)* 2006], a encriptação AES-128 e a linguagem de descrição de equipamentos DDL/EDDL. O *Wireless*HART é um tecnologia de comunicação segura que opera na frequência ISM(*Industrial, Scientific and Medical)* de 2.4GHz. O protocolo utiliza rádios DSSS (*Direct Sequence Spread Spectrum)* compatíveis com o padrão IEEE-802.15.4 e o FHSS (*Frequency Hop Spread Spectrum*) para o chavemento de canais pacote a pacote.

O *Wireless*HART dá suporte a um conjunto de dispositivos básicos, que seriam (Figura 2.2):

- Dispositivos de campo (*Field Devices*): Dispositivos básicos que realizam função de sensores ou atuadores.
- Roteadores (*Routers*): Servem para encaminhar os pacotes que trafegam na rede.
- Adaptadores (*Adapters*): Conectam um dispositivo HART legado (cabeado) à rede sem fio
- Dispositivo Portátil (*Handheld*): dispositivos móveis utilizados por usuários.
- Pontos de Acesso (*Access Points*): Conectam dispositivos de campo ao Gateway.
- Gateway: Funciona como um intermediário com a aplicação (pode haver redundância).
- *Network Manager*: Gerencia o escalonamento por divisão de tempo da rede.

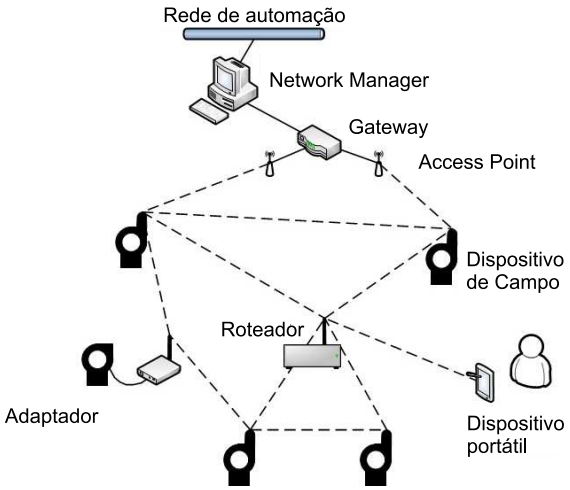


Figura 2.2: Dispositivos *Wireless*HART.

No protocolo *Wireless*HART, as mensagens são enviadas utilizando *Time Division Multiple Access* (TDMA). Em tal técnica, o tempo é dividido em pequenos intervalos chamados *slots*. O seu funcionamento opera de maneira que cada transmissão ocorrerá no *slot* de tempo atribuído a mesma. Com a sucessão dessas transmissões em intervalos de tempo se dá o fluxo da rede. Como cada dispositivo transmite em um espaço de tempo diferente as colisões dentro de um mesmo canal são evitadas. No *Wireless*HART o dispositivo responsável por essa distribuição de *slots* é o *Network Manager*, que é mais detalhado na seções seguintes.

A grande maioria das transmissões são direcionadas por algoritmos de rotas em grafos. O escalonamento é realizado de forma centralizada pelo *Network Manager*, baseado

nas informações de roteamento vindas de toda a rede juntamente com as necessidades de cada dispositivo e de cada aplicação. O escalonamento é subdividido em pequenos intervalos de tempo denominados *slots* e enviado a cada dispositivo pelo *Network Manager*. Cada dispositivo recebe *slots* de acordo com sua necessidade. O *Network Manager* é sensível a mudanças na topologia e na demanda de banda, realizando atualizações no grafo representativo da rede e no escalonamento dos *slots*.

2.1 Roteamento

O processo de roteamento consiste na seleção dos caminhos nos quais os dados são enviados em uma rede de comunicação. No protocolo *Wireless*HART todos os dispositivos devem ser capazes de encaminhar dados, além de produzir e/ou recebê-los para que mantenha-se a comunicação em malha. Logo, no *Wireless*HART, o roteamento deve definir por quais dispositivos os dados passarão, problema este simplificado porque todas as comunicações são realizadas através do *gateway*, não havendo comunicação fim-a-fim direta entre os dispositivos.

O roteamento é representado por três tipos de grafos, *broadcast*, *downlink* e *uplink*, ilustrados na Figura 2.3 (adptada de [Han et al. 2011*b*]).

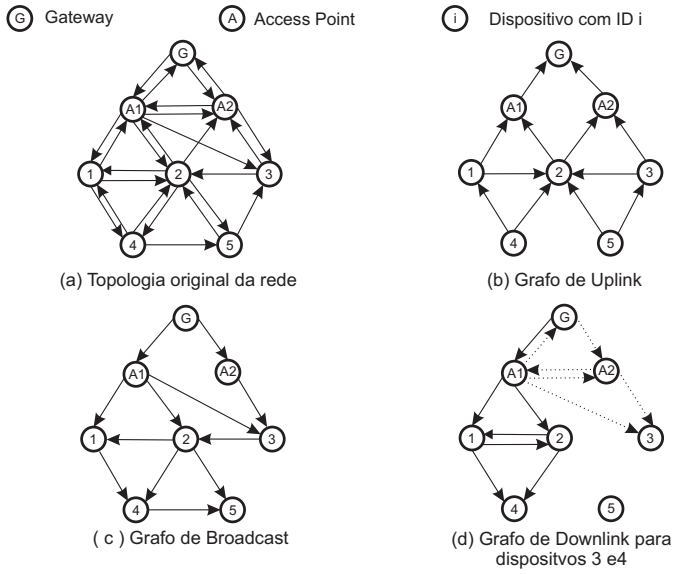


Figura 2.3: Tipos de grafos no *Wireless*HART.

O grafo de *broadcast* liga o o *gateway* a todos os dispositivos da rede. É utilizado para o envio de mensagens de configuração e de controles comuns para toda a rede a partir do *gateway*. O grafo de *uplink* é o que conecta cada um dos dispositivos até o *gateway* para

o envio periódico dos dados de processo. Finalmente o grafo de *downlink* é o grafo que liga individualmente o *gateway* a cada dispositivo. De modo que são enviadas mensagens de maneira unicast para cada dispositivo por este grafo.

Para manter a comunicação em malha, cada dispositivo *Wireless*HART deve ser capaz de encaminhar pacotes para outros dispositivos. Existem três formas de roteamento definidos no padrão *Wireless*HART [(IEC) 2010].

Roteamento Por Grafo: Um grafo, no contexto do *Wireless*HART, é um conjunto de caminhos que ligam os dispositivos da rede. No roteamento por grafos (*Graph Routing*), os caminhos em cada grafo são criados pelo *Network Manager* e copiados individualmente para cada dispositivo da rede. Para enviar um pacote, o dispositivo transmissor escreve um identificador de grafo (determinado pela estação de destino) no cabeçalho da mensagem. Todos os dispositivos da rede devem estar pré configurados com as informações sobre o grafo, especificando para quais vizinhos o pacote deverá ser encaminhado.

Roteamento de Origem: O roteamento de Origem (*Source Routing*) é uma extensão do roteamento por grafo objetivando diagnósticos de rede. Para enviar um pacote ao seu destino, o dispositivo de origem inclui no cabeçalho uma lista de dispositivos pelos quais o pacote deve passar. No decorrer do roteamento do pacote, cada dispositivo roteante utiliza o próximo endereço de dispositivo de rede na lista para determinar o próximo dispositivo, até que o destino seja alcançado.

Roteamento de Superframe: O roteamento de *superframe* (*Superframe Routing*) é um caso especial do roteamento por grafo. No roteamento por *superframe*, os pacotes são designados para um *superframe* determinado inserindo o identificador do *superframe* (*Superframe ID*) na mensagem. O *superframe* será construído pelo *Network Manager*. Cada dispositivo que esteja associado com qualquer *link* no *superframe* deve receber a informação sobre o *superframe* e o *link*. O dispositivo deverá selecionar o primeiro *link* livre para encaminhar a mensagem, independente de qual seja o vizinho. Qualquer dispositivo que transmita sua mensagem com roteamento de *superframe* deve saber o endereço do dispositivo associado ao *superframe*. Uma vez que o pacote segue o *superframe*, não se faz necessário a configuração explícita das arestas do grafo.

A camada de transporte dá suporte a transmissões com e sem confirmação de recebimento do destinatário (ACK). Vale salientar que no serviço sem confirmação os dados enviados podem ser recebidos fora da ordem em que foram enviados originalmente. O serviço com confirmação constrói um circuito virtual síncrono de transporte pela rede conectando dois dispositivos, permitindo-os enviar dados e confirmar sua entrega de maneira síncrona. Com isso, se evitam perdas, desordenação e duplicações nos dados. Apenas uma transação por circuito é permitida por vez. [(IEC) 2010]

O roteamento de toda a rede é determinado pelo *Network Manager*. Com o objetivo de construir rotas eficientes e otimizadas, o *Network Manager* precisa de informações sobre a rede, sobre os requisitos de comunicação e informações sobre a capacidade dos dispositivos de rede. Com a descoberta dessas informações, o *Network Manager* ajusta as conexões na rede para um melhor funcionamento da mesma. Os requisitos necessários ao *Network Manager* para o roteamento são mostrados na Tabela 2.1.

Tabela 2.1: Requisitos de roteamento.

Atividade da rede	Requisitos
Criar e gerenciar rota da rede	O *Network Manager* mantém uma representação interna da rede, que em casos extremos pode apresentar todos os nós interligado um a um. O *Network Manager* direciona a representação para o que ele estima ser uma representação racional da rede. A representação interna é utilizada para produzir o roteamento por grafo e de origem.
Gerenciar tabela de vizinhos	O *Network Manager* coleta dados estatísticos da rede e as tabelas de vizinhos dos dispositivos através de relatórios periódicos. Informações sobre as conexões existentes e sobre o nível de sinal são utilizadas para ajustar as rotas.
Relatórios	As informações sobre as comunicações são utilizadas para decidir sobre conexões existentes ou criar novas conexões.
Construir tabelas de roteamento para o roteamento por grafo	O roteamento por grafo é ideal para comunicações escalonadas de *upstream* e *downstream*. Comunicações de *upstream* incluem medidas de processo e alarme. Comunicações de *Downstream* englobam as mudaças de controle dos atuadores. Não devem existir rotas circulares.
Construir listas para o Roteamento de Origem	O *Network Manager* constrói as rotas de origem. Não devem existir rotas circulares.
Construir Grafo de Downstream	O *Network Manager* gera um *broadcast* de *downstream* e um grafo *unicast* do *gateway* para cada um dos dispositivos de rede.

Apesar de que a norma [(IEC) 2010] não define um algoritmo para o roteamento, ela descreve a estratégia a ser utilizada. Tal estratégia é mostrada na a seguir:

1. Se existe um caminho com um único *hop* para o *gateway*, este deverá ser usado.
2. O número mínimo de *hops* a serem considerados na construção do grafo é 2.
3. O máximo número de *hops* a serem considerados na configuração do grafo inicial é 4.

4. Para comparar um caminho de um *hop* com um caminho de dois *hops* deve-se calcular uma proporção entre as intensidades de sinal que formam o caminho de dois *hops* para depois comparar esta proporção com intensidade de sinal do caminho de um *hop*.
5. A mesma regra descrita em (4) para caminhos com 3 e 4 *hops*.
6. O limite de nível de sinal a ser utilizando quando construindo o grafo deve ser inicialmente de 50%. Se nenhum caminho for encontrado com esse valor, o limite pode ser reduzido a 0,75 do seu valor e a geração do grafo refeita. A recursão deve continuar por até quatro vezes. Caso nenhuma única rota seja possível, o nó deve ser considerado inalcançável.

2.2 Escalonamento de Mensagens

Uma vez definidas as rotas (roteamento definido), o *Network Manager* deve definir quais dispositivos irão transmitir e em qual instante de tempo isso ocorrerá. Além disso, devido ao *Wireless*HART possibilitar o uso de múltiplos canais, o *Network Manager* também deve especificar o canal (frequência) a ser utilizado para cada transmissão. Portanto, este processo de atribuir determinado tempo e um canal para que se realize a comunicação entre duas estações específicas de modo a se cumprir as comunicações fim-a-fim da rede é chamado escalonamento.

Para organizar o processo de escalonamento de mensagens, o protocolo *Wireless*HART define a estrutura do *superframe*. Nas próximas subseções serão explicados os conceitos do *superframe* e do roteamento no protocolo.

2.2.1 Superframe

Uma característica importante do *Wireless*HART é a sincronização por tempo da camada de enlace, que define *slots* fixos de 10 ms e utiliza a tecnologia TDMA para fornecer comunicações determinísticas e sem colisões. O conceito de *superframe* é definido como um grupo de *slots* consecutivos. Um *superframe* é periódico, com o período sendo igual ao tamanho total dos *slots* membros. Todos os *superframes* em uma rede *Wireless*HART começam do ASN (*absolute slot number*) igual a 0, o tempo em que a rede foi primeiramente criada. Então, cada *superframe* se repete ao longo do tempo baseado no seu tamanho, definido o seu período. Quando o *superframe* é criado, este é associado a um *Graph_ID*. O *Network Manager* utiliza esta associação para alocar *links* nos *slots* de tempo. O *superframe* é uma combinação dos canais e dos *slots* de tempo. Os dispositivos

têm a disposição *links* para usar em tempo de execução.

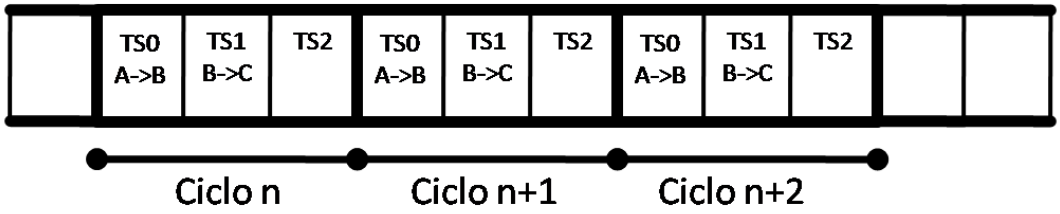


Figura 2.4: Ciclo de um *superframe* simples de 3 *slots*.

Cada nova instância de um *superframe* num determinado momento é chamada de ciclo do *superframe*. A Figura 2.4 mostra como os dispositivos se comunicariam em um *superframe* simples de 3 *slots*. No caso, o dispositivo A se comunica com o B no *slot* 0, o dispositivo B se comunica com o C no *slot* 1 e o *slot* 2 não é utilizado. O escalonamento se repete a cada 3 *slots*.

O tamanho dos *superframes* deve seguir uma cadeia harmônica [(IEC) 2010], de modo que todos os períodos podem ser divididos entre si. Exemplos de cadeias harmônicas seriam 1, 2, 4, 8, 16... e 3, 6, 12, 24... e também qualquer outro período que esteja de acordo com a expressão ab^n, onde a e b são dois números constantes e n é um número natural qualquer.

Uma rede *Wireless*HART pode conter vários superframes de diversos tamanhos dispostos em paralelo no tempo. Múltiplos *superframes* podem ser utilizados para determinar como os diferentes grupos de dispositivos vão se comunicar ou para que a rede funcione com diferentes ciclos de trabalho. *Superframes* adicionais podem ser criados para atender a diferentes freqüências de transmissões, requisitos de publicação de dados, notificação de eventos e comandos da camada de aplicação. O tamanho do *superframe* deve ser maior que o número de canais ativos de modo que um *link* em um *superframe* tenha a chance utilizado em qualquer um dos canais ativos.

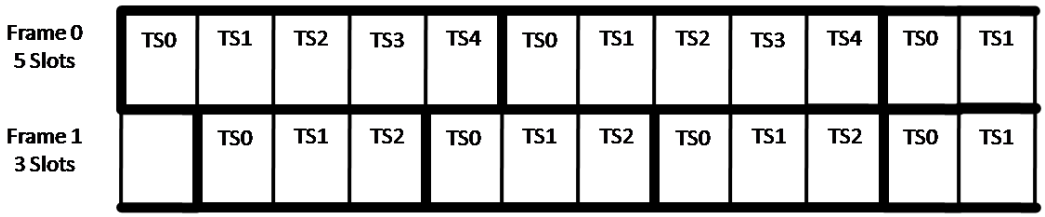


Figura 2.5: Relação entre múltiplos *superframes*.

Um dispositivo de rede pode participar em mais de um superframe simultaneamente, mas nem todos os dispositivos devem participar em todos os *superframes*. Configurando

um dispositivo para participar em múltiplos *superframes* de tamanhos diferentes e que se sobrepõem, possibilita estabelecer diferentes escalas de comunicação e matrizes de conectividade que funcionam simultaneamente.

Aplicações como sistemas de gerenciamento de ativos e aplicações específicas de dispositivos, geralmente requerem a transmissão de um volume de dados considerável durante curtos intervalos de tempo (medidos em minutos), que geralmente correspondem a chamadas de configuração, diagnósticos e respostas a requisições de usuários. Para que esta demanda seja suportada, *superframes* adicionais podem ser usados.

Superframes podem ser adicionados, removidos, ativados e desativados durante o funcionamento da rede. Como todos os *superframes* têm o mesmo tempo de início (ASN 0), o ciclo 0, *slot* 0 de todo superframe ocorre no início de uma época. A época para uma determinada rede é o tempo no qual o *Network Manager* inicializou a rede. Por causa disso, diferente *slots* de tempo em diferentes *superframes* estão sempre alinhados, mesmo que o inícios e os finais dos *superframes* estejam alinhados como na Figura 2.5. Como todos os *superframes* começam com o mesmo tempo, é sempre possível saber qual o tempo de início de um ciclo do superframe e de um *slot* de tempo.

2.2.2 Princípios do Algoritmo de Escalonamento de Mensagens

O escalonamento de mensagens no *Wireless*HART é realizado centralizadamente pelo *Network Manager*. Para que o algoritmo de escalonamento seja feito de maneira eficiente e otimizada, o *Network Manager* precisa de informações sobre a rede, sobre os requisitos de comunicação e sobre a capacidade dos dispositivos na rede. Ao passo que estas informações são obtidas e atualizadas, o *Network Manager* continua a ajustar o escalonamento para que ele se adeque aos requisitos. Assim, o escalonador utiliza informações do sistema operacional do dispositivo para ajustar o escalonamento. Os requisitos de escalonamento de mensagens definidos na norma [(IEC) 2010] são descritos na Tabela 2.2.

A norma *Wireless*HART [(IEC) 2010] não especifica um algoritmo de escalonamento de mensagens, mas resume a seguinte estratégia escalonamento:

1. Estratégia de escalonamento
    - Iniciando do *slot* 0, o *channel offset* é atribuído aos dispositivos .
    - O dispositivo que publica dados mais rapidamente é alocado primeiro.
    - O destino dos dados publicados é sempre o *gateway*.
2. Superframes de Dados

- O comprimento do superframe de dados é determinado pela taxa de leitura de dados.
- Os *slots* são alocados da mais rápida para mais lenta taxa de atualização de dados.
- Se iniciando do dispositivo mais distante do *gateway*, um *link* é alocado para cada dispositivo na rota para o *gateway*. Um segundo *slot* é alocado para dar suporte a retransmissão.
- cada transmissão é escalonada com uma retransmissão alocada em outro caminho (caso exista um disponível).
- Um dispositivo de rede só pode estar escalonado para receber dados uma vez por *slot*.
- Notificações de eventos utilizam o mesmo esquema de escalonamento dos dados. Se houver uma operação de publicação de dados escalonada, então os eventos podem dividir o mesmo *slot*. Os eventos são enviados esporadicamente. Quando um evento é enviado, pode-se usar o *slot* de retransmissão.

3. Superframe de Gerenciamento (*Management Superframe*)

   - O superframe de gerenciamento tem prioridade sobre os *superframes* de dados.
   - O superframe do *Network Manager* deve compor-se de 6400 *slots* conforme descrição da norma [(IEC) 2010].
   - O grafo deve ser percorrido por busca em largura, iniciando do *gateway*, numerando os dispositivos de N0, N1, ... Nn.
   - Todo dispositivo deve ter um *slot* para um DLPDU de *Keep-Alive*. Este *slot* deve ser um *slot* de recepção compartilhado do nó pai no grafo compartilhado. Se um dispositivo de rede não envia nenhum pacote para o seu nó pai durante o intervalo de tempo denominado *Keep_alive_time*, deve enviar um DLPDU de *Keep-Alive* depois que este tempo expirar.
   - Cada dispositivo deve ter 3 *slots* a cada 15 minutos para relatórios de estado do dispositivo.
   - Cada dispositivo deve ter pelo menos 1 *slot* compartilhado a cada minuto para requisição/resposta. Caso esteja apto, o *gateway* deve alocar os recursos necessários para esta aplicação.

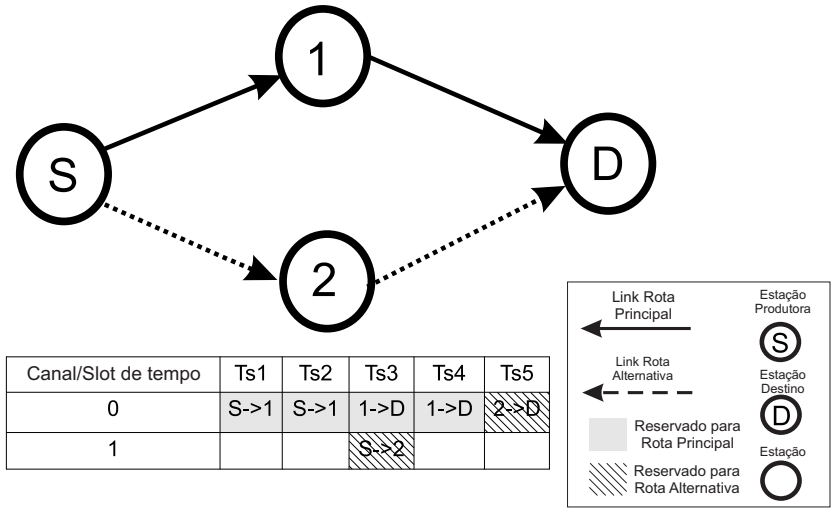
4. Comandos e respostas para gerenciamento da rede

   - Os *links* de gerenciamento de rede devem ser compartilhados com requisição e respostas de *join*.

5. Comandos requisição/respostas
    - Os *links* devem ser alocados de mesmo modo das requsições de *join*.
    - O *Network Manager* aloca *slots* compartilhados para suportar o tráfico *ad-hoc* de requisição/respostas.
6. Superframe de Gerenciamento
    - Devem existir *slots* alocados para o superframe de gerenciamento. Este superframe deve estar configurado em todos os dispositivos. Ele deve ter 1 segundo de duração e ser composto de 4 *slots*.
7. Superframe de Gateway
    - O superframe de *gateway* deve ter Superframe_ID de valor 253.
    - O superframe de *gateway* tem, no mínimo, 40 *slots* de duração (400ms).
    - Os *slots* devem ser alternados entre *slots* de transmissão e de recepção e devem todos serem compartilhados.
8. Superframe de Propósitos Especiais
    - O *Network Manager* pode alocar *superframes* para serem usados pelo *gateway* ou por um cliente para atender altas demandas de transmissão de dados. Isso é definido com serviço de "manutenção"ou "tranferência em bloco".
    - O *Network Manager* deve alocar *superframes* para serem usados pelo dispositivos *handheld* e por todos os dispositivos de campo para propósitos de manutenção. Este superframe é usado para fornecer conexões temporárias para *handhelds* e dispositivos de campo. O *Network Manager* deve alocar 4 *slots* por segundo (dois *links* em cada direção).
9. Parâmetros para otimização
    - Número de *hops* ao *gateway*.
    - Caminhos alternativos.
    - Latência.
    - Consumo de energia
    - Volume total de dados transmitidos.

Pela sugestão de algoritmo da especificação uma rota secundária pode ser utilizada, além de uma rota principal que liga a origem dos dados ao destino. Para cada pacote a ser transmitido por um dispositivo, dois *slots* de tempo devem ser escalonados: o primeiro para a transmissão e o segundo para uma possível retransmissão (ambos pela rota principal). Caso uma rota alternativa esteja definida para este *link*, uma terceira cópia do pacote

deve ser enviado por tal rota. Um exemplo simples de escalonamento pode ser visto na Figura 2.6.



Canal/Slot de tempo	Ts1	Ts2	Ts3	Ts4	Ts5
0	S->1	S->1	1->D	1->D	2->D
1			S->2		

Figura 2.6: Um exemplo de escalonamento em uma topologia simples.

No primeiro *slot* de tempo do exemplo descrito na Figura 2.6, o dado é produzido na estação fonte 'S' e enviado para o estação de destino 'D' utilizando a rota principal S-1-D e a rota alternativa S-2-D. De acordo com o algoritmo de escalonamento previamente descrito, o primeiro e o segundo *slot* de tempo, Ts1 e Ts2, no *offset* de canal 0 estão reservados para transmissão e retransmissão do pacote pela rota principal. Ts3 Reserva o *offset* de canal 1 para a transmissão pela rota alternativa. Novamente, o *offset* de canal 0 em Ts3 e Ts4 estão reservados para o transmissão e retransmissão na rota principal. Finalmente em Ts5, o *offset* de canal 0 é reservado para a transmissão final do último *hop* pela rota alternativa.

Uma retransmissão para melhorar a confiabilidade sendo realizada antecipadamente, leva a uma limitação na banda mesmo quando retransmissões forem desnecessárias. Entretanto, o *Wireless*HART provê recursos para superar esta limitação [Chen et al. 2010].

2.3 Health Report

Na rede *Wireless*HART, o *Network Manager* é o centralizador das atividades de gerenciamento (por exemplo roteamento e escalonamento). Para que isso seja possível, é necessário que o *Network Manager* tenha acesso a dados dos outros dispositivos da rede. O mecanismo que a [(IEC) 2010] impõe é a utilização de relatórios periódicos chamados *Health Reports* que são enviados ao dispositivo gerenciador da rede. Esta estrutura tem

importância para a presente tese uma vez que os dados reportados servem como base para realização dos processos de roteamento e escalonamento.

Segundo [Chen et al. 2010], a habilidade de um dispositivo em se comunicar com os vizinhos é essencial para a formação e o funcionamento de uma rede em malha. Consequentemente, os dados para tal comunicação são mantidos nas tabelas de vizinhos. Tais dados incluem a intensidade de sinal média recebida (Received Signal Level - RSL), estatísticas sobre o envio e o recebimento de pacotes e o registro de tempo da última transmissão realizada com o vizinho.

Periodicamente, um dispositivo envia um *health report* sobre seus vizinhos para que o *Network Manager* atualize a situação dos dispositivos e seus vizinhos. O valor de RSL é restabelecido a cada *health report*, sendo importante para a otimização da configuração da rede. Até quatro vizinhos são representados por vez nas mensagem, contudo, o *Network Manager* pode requisitar um maior número. Existem dois comandos para se produzir os *health reports* (a informação de tais comandos é exclusiva e devem ser reportadas pelos dispositivos periodicamente). O Comando 780 fornece as estatísticas dos vizinhos, incluindo as estatísticas de comunicação. O comando 787 provê o nível de sinal dos nós vizinhos descobertos (não os vizinhos já ligados ao nó). Um dispositivo pode descobrir um vizinho através de um *link* de descoberta.

Novos dispositivos podem descobrir seus vizinhos através do recebimento de mensagens de *advertise*. De fato, a requisição de entrada na rede de um novo dispositivo inclui os dados de um comando 787. Caso necessário, no processo de entrada na rede, o dispositivo pode envia mensagens de *keep alive* no *link* de entrada para se manter sincronizado e coletar dados .

Tabela 2.2: Requisitos de escalonamento de mensagens.

Função de Rede	Requisito
Pressupostos	O *Network Manager* tem uma representação do grafo da rede. Cada dispositivo de rede deve ser configurado com uma tabela de conexões. O network manager sabe a taxa de atualização de cada dispositivo. Para efeito de redundância, dados são enviados com uma transmissão e uma retransmissão por um caminho e outra retransmissão em um caminho diferente.
Restrições	-O número máximo de canais ativos é determinado pelo número de canais habilitados e limitado pelo *blacklisting*. -Nenhum dispositivo deve ser escalonado para escutar duas vezes em um mesmo *slot*, contudo, mais de um dispositivo pode transmitir para o mesmo dispositivo (*slot* compartilhado). -Um *link* de *broadcast* e *links* dedicados para cada um dos dispositivos que escutam a transmissão podem coexistir. -Em um caminho de múltiplos *hops*, o *hop* mais imediato é escalonado primeiro. -As taxas de atualização suportadas são definidas por 2^n, onde n é um valor inteiro. Por exemplo, a taxa de atualização aceita pode ser 250ms, 500ms, 1s, 2s, 4s, 8s, 16s, 32s, 64s ou mais. -Gerenciamento básico da rede e comunicação de dados publicados (*publish data*) não devem exceder 30% da banda disponibilizada (100*slots*/seg no máximo). -O *Network Manager* leva em conta os requisitos e serviço. -O escalonamento final (não contando o *gateway*) deve ter 50% de *slots* livres (ou seja, alocados para retransmissões e escutas).
Descoberta de vizinhos	O *Network Manager* aloca um *link* de descoberta comum a todos os dispositivos de rede. Um temporizador de descoberta é configurado para controlar esse processo.

Capítulo 3

Estado da Arte

Neste capítulo apresentaremos uma revisão da literatura das principais soluções para o roteamento e para o escalonamento de mensagens em redes *Wireless*HART. São enfatizados algoritmos de roteamento e escalonamento de mensagens que estão em conformidade com a norma *Wireless*HART. O Capítulo foca primeiro nos algoritmos de roteamento de mensagens, seguido pelos algoritmos de escalonamento de mensagens e por fim apresenta os principais simuladores disponíveis para a tecnologia *Wireless*HART.

3.1 Estado da arte dos esquemas de roteamento de mensagens para *Wireless*HART

Uma topologia de rede é definida por um conjunto de dispositivos e suas conexões entre si. Entretanto, no caso de redes sem fio, devido a atenuação e a interferência, muitos destas conexões (*links*) não podem ser usadas. Com isso, no contexto das RISSF, o roteamento consiste em selecionar os *links* a serem utilizados dentre os disponíveis em uma topologia.

Roteamento é muito importante para o tempo de vida da rede e o seu desempenho, uma vez que define por quais dispositivos os dados irão trafegar, afetando *delay*, o consumo de energia e a taxa de transferência de dados por exemplo [Silva et al. 2010*b*].

Nesta seção serão apresentados as mais relevantes propostas de algoritmos de roteamento para a tecnologia *Wireless*HART, destacando as principais características dos mesmos.

Uma avaliação prévia dos algoritmos de roteamento para *Wireless*HART foi realizado em [Dickow 2014], o qual selecionou três algoritmos (o algoritmo de roteamento Han [Han et al. 2011*b*], o Bellman–Ford [Cormen et al. 2005], e o ELHFR [Jindong et al. 2009]) para testar sua aplicabilidade ao *Wireless*HART. A partir de uma análise de

grafos, o autor concluiu que os algoritmos Han e Bellman–Ford são adequados a redes *Wireless*HART e tiveram desempenhos simulares nos testes (com uma leve vantagem para o algoritmo Han). O ELHFR teve uma desempenho inferior aos outros. Por esta razão apenas o algoritmo Han será utilizado no capítulo de resultados entre os três nesta seção.

3.1.1 Algoritmo de roteamento Han

Em [Han et al. 2011*b*] é apresentada uma solução completa para o roteamento e o escalonamento de mensagens em redes sem fio em malha. Em [Zand et al. 2012], são apresentadas as implementações desses algoritmos. Nesta seção apenas o algoritmo de roteamento será apresentado. Os autores em [Han et al. 2011*b*] afirmam que a maioria dos algoritmos apresentados em [Mueller et al. 2004], [Tarique et al. 2009], [Ganesan et al. 2001], [Lee & Gerla 2001], [Marina & Das 2001] e [Ye et al. 2003] focam em identificar caminhos de múltiplos nós ou nós disjuntos para melhorar a confiabilidade do roteamento. Contudo, o *Wireless*HART impõe requisitos de confiabilidade mais restritos para lidar com o ruidoso e severo ambiente industrial. Por essa razão, a literatura apresentada no trabalho não pode ser diretamente aplicada as redes *Wireless*HART, e novos algoritmos de roteamento tem de ser desenvolvidos.

Como método de validação, em [Han et al. 2011*b*] um sistema de comunicação *Wireless*HART completo foi implementado e as propostas de solução foram implementadas no *Network Manager*. Contudo os autores abstraem alguns requisitos de roteamento apresentados na especificação do *Wireless*HART. Esta abstração é feita para que os autores possam utilizar a sua definição de (k,m)-reliability, o qual define o número de arestas que entram/saem de cada nó da rede. Além disso, os autores assumem que o *gateway* tem uma conexão cabeada e confiável, sendo os requisitos de confiabilidade aplicados apenas aos dispositivos sem fio.

A métrica escolhida para o algoritmo guloso foi o número médio de saltos (h - do inglês, *hops*), o qual é calculado com base no número de saltos até os nós pai. O h do nó filho é a média dos menores h dentre os nós pai acrescido de 1. Primeiro, o algoritmo constrói um grafo de *broadcast* adicionando um nó não visitado por iteração ao "grupo de nós confiáveis"("V_B") baseado no número de arestas vindas dos nós candidatos para V_B. A preferência é por nós com pelo menos duas arestas para V_B, e então são adicionados o nó e as duas arestas que vem de pais com o menor h para o grafo de *broadcast* (G_B). Então, aqueles com apenas uma aresta para V_B são adicionados.

Após a construção de G_B, o algoritmo constrói o grafo de *uplink* (G_U) invertendo a direção de todas as arestas em G_B. De tal modo, se constrói uma árvore que conecta todos

os nós ao *gateway*.

Para o caso do *downlinks*, o trabalho começa definindo um algoritmo que cria um grafo de *downlinks* para cada dispositivo na rede, assim como define a especificação do *Wireless*HART. Essa abordagem se torna mais complicada (devido a presença de ciclos) e não é escalável, uma vez que o grafo inclui múltiplos nós intermediários (o que gera *overhead*) mas não pode reutilizar a informação do grafo de *downlinks*. Com isso, o trabalho propõe o Roteamento de *Downlinks* Sequencial Confiável (*Sequential Reliable Downlink Routing* - SRDR), o qual transforma a rota de *downlinks* em uma sequência de rotas locais. Os grafos locais de nós intermediários servem como partes para a construção de rotas mais longas. Para que estes algoritmos sejam implemetados, duas extensões ao protocolo *Wireless*HART são necessárias. Primeiro, o algoritmo deve utilizar os bits reservados (bits de 4–3) do byte de controle do cabeçalho da camada de rede para indicar a presença dos campos de sequencia do roteamento de *downlinks*, armazenando a lista do grafo ordenado no campo de opção de *source routing*. Segundo, deve-se especificar como o módulo de roteamento lida com o SRDR, verificando, por exemplo, quando o um dado nó é parte de um dos grafos intermediários até o destino do pacote e enviando relatórios adequados ao *Network Manager*.

3.1.2 Algoritmo de Roteamento Conjunto para Maximização do Tempo de Vida da Rede (JRMNL)

O Algoritmo de roteamento conjunto para Maximização do Tempo de Vida da Rede (em inglês *Joint Routing Algorithm for Maximizing Network Lifetime* - JRMNL) foi proposto por [Zhang, Yan & Ma 2013]. Este algoritmo é baseado no roteamento por grafo (*graph routing*) do WirelssHART e define a função custo para a seleção do próximo nó com o objetivo de maximizar o tempo de vida da rede. Antes de aplicar a função custo, o algoritmo atribui graus aos nós com base no número de saltos entre o nó e o *gateway* (o *gateway* tem grau 0). Então, os *links* entre nós de um mesmo grau são removidos, como pode ser visto na Figura 3.1.

A função custo para o *link* entre os nós i e j são definidos na equação 3.1, e são baseadas no fator de carga de comunicação do nó (B_{j1}), na energia consumida para a transmissão do nó i para o nó j ($P_d(i,j)$) e energia inicial e final do destino (E_o e E_{kj}, respectivamente). O ω_i representa o conjunto de nós vizinhos do nó i. As variáveis x_1, x_2 e x_3 são inteiros positivos que definem a prioridade de cada característica.

$$L_{i\to j} = min_{l\in\omega_i}((B_{j1})^{x_1} P_d(i,j)^{x_2} (\frac{E_0}{E_kj})^{x_3}) \tag{3.1}$$

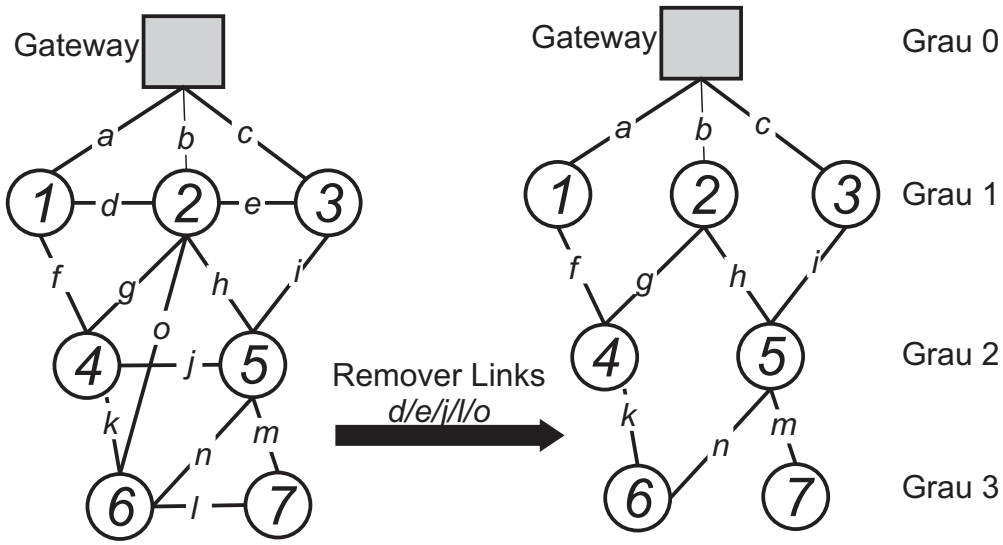


Figura 3.1: Adaptação da topologia do algoritmo JRMNL.

O fator de carga de comunicação é definido por [Zhang, Yan & Ma 2013] e é baseado no intervalo de transmissão médio entre cada nó e seus vizinhos. O $P_d(i,j)$ é definido pela Equação 3.2, onde α é o expoente de atenuação, d_{ij} é a distância linear entre os nós i e j, R_0 é a taxa de transmissão, e P_0^s é a probabilidade de uma transmissão ser bem sucedida. Os valores referentes ao ruído são definidos por uma Variável Aleatória Gaussiana Complexa de média zero com variância N_0. Esta equação foi desenvolvida por [Ibrahim et al. 2008].

$$P_d(i,j) = \frac{(2^{R_0}-1)N_0 d_{ij}^{\alpha}}{-log(P_0^s)} \tag{3.2}$$

Como pode ser visto, a função custo é incrementada por B_{j1} e $P_d(i,j)$ mas é decrementada por E_{kj}. Isso significa que o melhor caso do algoritmo para um nó é aquele no qual a carga de comunicação e e a energia de transmissão são mínimas e a energia residual é máxima.

O algoritmo foi validado utilizando uma simulação em Matlab® e foi comparada com os algortimos ELHFR [Jindong et al. 2009] e MPCR [Ibrahim et al. 2008], obtendo uma melhora no tempo de vida da rede de 7 vezes e 2 vezes, respectivamente. Para o potÊncia

média de transmissão, o JRMNL teve aproximadamente 4 dBm a menos que o ELHFR e 2 dBm a mais que o MPCR.

3.1.3 Algoritmo de roteamento Re-Add

O Re-add é um algoritmo de roteamento proposto por [Zhang et al. 2014] que aplica uma função de prioridade ponderada em conjunto com o algoritmo de construção do do grafo de *uplink* para determinar as rotas da rede. Partindo da topologia original, o algoritmo atribui graus (quantidade de saltos até o *gateway*) para cada nó e remove *links* entre nós do mesmo grau, assim como o JRMNL. Entretanto, em casos nos quais o nó possua apenas um *link* para os nós de menor grau, o algoritmo re-adiciona *links* entre nós com mesmo grau para que se melhore a confiabilidade nestes casos. O *link* re-adicionado deve ser o do nó mais próximo para que se garanta o menor consumo de energia na transmissão. Este processo melhora a confiabilidade da rede porque ele provê cada nó com dois possíveis caminhos (caso existam) para que os pacotes sejam encaminhados. O processo é ilustrado na Figura 3.2.

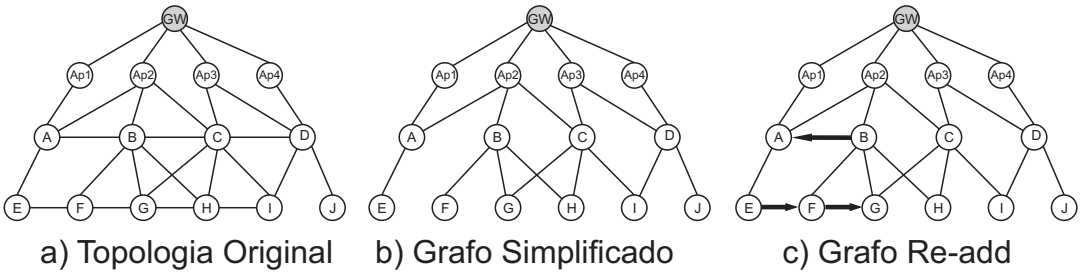


Figura 3.2: Topologia original do grafo re-adicionado.

Além disso, o algoritmo utiliza a Equação 3.3 para atribuir a prioridade do *link* entre o nó i e o nó j (P_{ij}). A equação usa os seguintes critérios: Qualidade do *Link* (Q_{ij}), energia residual no nó e a diferença entre os graus do nó i e o nó j (ou seja, $D_i - D_j$).

$$P_{ij} = x_1 Q_{ij} + x_2 \frac{E_i + E_j - P_d(i,j)}{2E_0} + x_3 \frac{D_i - D_j}{D_i} \tag{3.3}$$

Note que $P_d(i,j)$ é o mesmo consumo de energia da transmissão definido pela Equação 3.2 e utilizado pelo algoritmo JRMNL. Aqui, Q é a relação entre o número de pacotes recebidos e o número de pacotes transmistidos. Entretanto, para que se reduzam os custos com o mensagens de controle para avaliação de *link*, [Woo et al. 2003] propõe um estimador que utiliza uma janela de média com média móvel exponencialmente ponderada

(*window mean with exponentially weighted moving average* - WMEWMA) e a qualidade do *link* é calculada pela Equação 3.4.

$$\widehat{Q}_{k+1} = m\widehat{Q}_k + (1-m)\frac{recebido}{recebidos + falhas} \tag{3.4}$$

$\widehat{Q}_{k+1}$ e $\widehat{Q}_k$ representam a qualidade do *link* nos momentos k e $k+1$. Os valores *recebidos* e *falhas* representam o número de pacotes recebidos e perdidos por um nó em uma janela de tempo t. Além disso, o algoritmo utiliza $m \in [0,1]$ para controlar o histórico do estimador.

Finalmente, x_1, x_2 e x_3 são pesos que definem a relevância de cada critério (qualidade, energia e grau, respectivamente). O incremento em qualquer desses valores representa um incremento na prioridade do *link*. O algoritmo utiliza o processo analítico hierárquico definido em [Saaty 1980] para determinar os valores dos pesos. O processo de definição dos pesos inicia com uma matriz definida pelo usuário que realiza uma comparação item-a-item dos critérios de otimização, determinando quando cada um deles é mais, igual ou menos importante que cada um dos outros. Então uma matriz de julgamento é calculada (ver [Zhang et al. 2014] para detalhes) e os pesos são calculados por uma checagem de consistência.

O autor afirma que foram utilizadas simulações para validação, comparando o algoritmo com os algoritmos de roteamento Han [Han et al. 2011*b*] e o JRMNL [Zhang, Yan & Ma 2013]. Entretanto, o artigo apenas fornece os parâmetros de simulação e detalhes do experimento. Nenhuma informação sobre o simulador é fornecida. Os resultados sugerem que o Re-add supera os outros algoritmos em termos de confiabilidade e tempo de vida da rede. Maiores detalhes em [Zhang et al. 2014].

3.1.4 Algoritmos de roteamento não-*Wireless*HART

Para prover algumas sugestões e ideias para o desenvolvimento de algoritmos de roteamento *Wireless*HART, esta seção apresenta alguns algoritmos de roteamento que não foram desenvolvidos especificamente para o *Wireless*HART. As ideias apresentadas podem servir de guia para novos algoritmos *Wireless*HART.

Para a tecnologia ISA100.11a, [Pham & Kim 2013] e [Pham & Kim 2014] aplicam um modelo matemático de Taxa de Recepção de Pacotes (*Packet Reception Rate*-PPR) para definir regiões de conectividade; tais regiões são classificadas pela taxa de sucesso de suas rotas. Dois grafos são construídos: o primeiro é baseado no consumo de energia dos nós, o segundo é baseado em latência. Então o algoritmo de escalonamento é aplicado

nos dois grafos para que se mantenha o equilibrio de prioridades entre o consumo de energia e a latência. Os mesmos autores também produziram em [Quang & Kim 2014] um algoritmo de roteamento para Redes de Sensores de Rádio Cognitivo com base no ISA100.11a. O algoritmo estima a vazão máxima de dados para cada caminho e envia dados pelo caminho mais otimizado.

O algoritmo apresentado em [Quang & Kim 2014] utiliza uma estrutura baseada em *clusters* no qual os dispositivos (chamados de clusters) são responsáveis por sentir e alocar os canais de transmissão para os dispostivos vizinhos. Um *Cluster Head* é um nó especial que utiliza dois transmissores de rádio separados ao mesmo tempo, permitindo que o tempo ocioso do dos canais ISM licenciados sejam utilizados. Cada cluster pode estimar o estado do meio de transmissão utilizando o método de janela de média com média móvel exponencialmente ponderada (*window mean with exponentially weighted moving average*.

O protocolo *Hydro*, apresentado em [Dawson-Haggerty et al. 2010], tem o objetivo de reduzir o *overhead* de transmissões p2p (per-to-peer). O Hydro utiliza a métrica ETX [De Couto et al. 2003] para medir a qualidade do *link*, e adota três primitivas para para atingir os objetivos propostos. A primeira primitiva determina a formação de um Grafo Direcionado Acíclico (*Directed Acyclic Graph*–DAG), então cada nó deve manter uma lista de dispositivos de roteamento padrão que encaminhem os pacotes para os roteadores de borda. A segunda primitiva é construção de topologia global que define uma mensagem chamada *Topology Report* (Relatório de Topologia) que são anexados aos dados enviados. Este relatório contém informações sobre os melhores vizinhos na lista de rotas padrão, de modo que um grafo possa ser construído. E, finalmente, existe a instalação centralizada de rotas, a qual detemina quando as rotas devem ser instaladas/atualizadas nos dispositivos pelos roteadores de borda utilizando um mensagem de instalação de rota.

O trabalho em [Yu et al. 2014] propõe o protocolo REALFLOW, que tem enfoque específico em RISSF. Apesar de que o trabalho apresentado em [Yu et al. 2014] seja interessante devido a seu foco nos desafios das RISSFs, ele não está em confirmidade com a atual norma *Wireless*HART, se tornando mais como uma opção para futuras atualizações na tecnologia. Contudo esse trabalho foi contemplado nessa subseção devido à sua proposta inovadora. O protocolo inclui um algoritmo de roteamento, o qual provê diversidade de multiplos caminhos (melhora da confiabilidade), desempenho para dados em tempo real, tolerância a mudanças súbitas de topologia e uma divisão da carga de trabalho entre os dispositivos, minimizando o gasto de tempo de processamento com calculos de informações de roteamento. Para se atingir os objetivos propostos anteriormente, os autores dividem a solução em duas partes: a parte de estabelecimento e gerência de rotea-

mento e parte de encaminhamento de pacotes. O funcionamento destas partes e a estrutura da rede são mostradas na Figura 3.3 (adaptada de [Yu et al. 2014]).

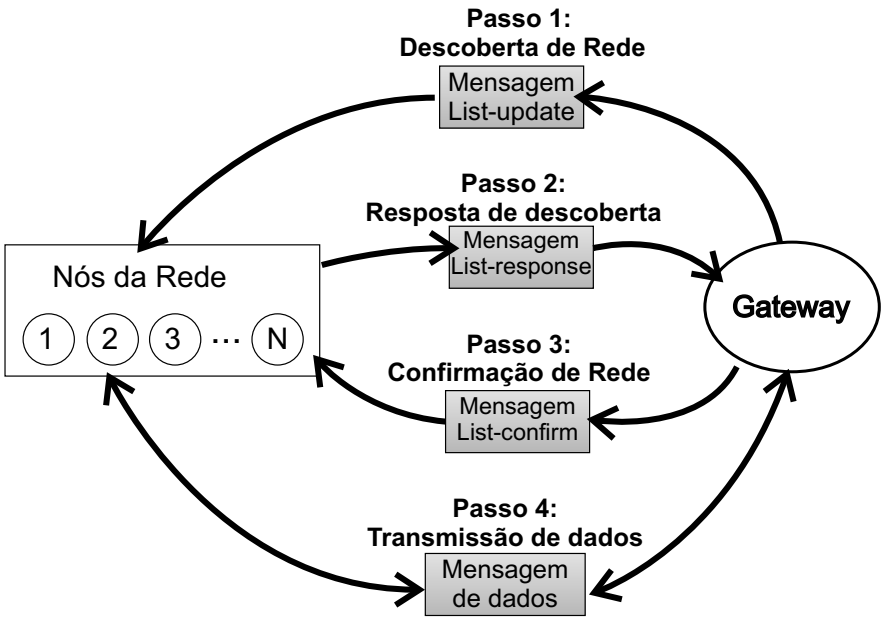


Figura 3.3: Resumo de funcionamento do REALFLOW.

Os passos 1, 2, e 3 apresentados na Figura 3.3 representam as três mensagens de controle usadas para o estabelecimento e manutenção do roteamento. A descoberta de rede é realizada pela mensagem *List-update*, a qual é enviada em *broadcast* do *gateway* para os dispositivos levando informações sobre a intesidade de sinais, número de saltos e o endereço do nó anterior. Quando um dispositivo recebe uma mensagem *List-update*, este dispositivo atualiza os parâmetros contidos na mensagem e a encaminha em *broadcast* de acordo com a necessidade para que chegue toda a rede. O acúmulo dos níveis de intensidade sinais determinam as rotas mais confiáveis, o número de saltos evita reenvios sem fim (o pacote é descartado se a contagem de saltos for maior que a profundidade do nó) e o endereço do nó anterior mantém o registro do caminho percorrido por cada pacote. Após receber as mensagens de *List-update*, os nós da rede enviam um relatório para o *gateway* através de uma mensagem *List-response*. Esta mensagem contém informação sobre os nós vizinhos (irmãos, pais e filhos) e sobre o próximo dispositivo para onde encaminhar pacotes a partir do dispositivo atual. O *gateway* utiliza estas informações para para construir a topologia da rede. Cada mensagem de *List-update* contém um número de sequência que evita a duplicação de pacotes. Finalmente, o *gateway* envia mensagens *list-confirm* para todos os nós com a confirmação das listas de nós relacionados e decisões de escalonamento. A supracitada parte de encaminhamento de pacotes consiste em um

conjunto de critérios que restringem a inundação de mensagens na rede, poupando recursos. O primeiro é a definição para cada nó n de uma lista local de nós relacionados L_n, a qual contém uma lista de endereços dos nós considerados relacionados a n. Um nó x é considerado relacionado a um nó n se x está envolvido no encaminhamento de pacotes entre n e o *gateway*. Além disso, o critério define que uma vez que o nó n recebe um pacote, o nó só irá encaminhar tal pacote se o endereço de origem ou o endereço do destino do pacote entejam contidos em L_n.

A segunda parte consiste em marcar os pacotes com um identificador único. Este identificador encapsula o número de sequência do pacote e o endereço de origem do nó atual. Os identificadores de pacotes são armazenados em cada nó utilizando uma tabela de histórico H, e uma vez que um novo pacote é recebido, o identificador que chega é comparado aqueles contidos em H. Se o identificador estiver presente em H, o pacote é uma duplicata e deve ser tratado apropriadamente como tal. Finalmente, o último critério é a idade do pacote. Pacotes desatualizados são de uso limitados em redes industriais, então cada pacote carrega um valor de idade. Este valor de idade é comparado com o prazo de entrega ao *gateway* para aquele pacote antes de ser enviado. Caso a idade seja maior que o prazo de entrega ao *gateway* o pacote é descartado para que se evite a utilização de *buffers* desnecessariamente em nós intermediários.

O incoveniente do algoritmo de roteamento apresentado pelos autores em [Yu et al. 2014] é o aumento no consumo de energia devido ao uso do algortimo de inundação e o foco em confiabilidade e em desempenho em tempo real.

3.1.5 Análise comparativa entre algoritmos de roteamento

Para ressaltar as informações relevantes sobre as propostas de algoritmos, nós discutiremos nessa seção os itens descritos abaixo. Além disso, a Tabela 3.1 apresenta uma visão das propostas analisadas.

- **Objetivo**: Os diversos algoritmos de roteamento são desenvolvidos para resolver diferentes tipos de problemas ou para melhorar características específicas na rede. Este item mostra o objetivo que os autores planejaram alcançar ao desenvolver a proposta de algoritmo.
- **Métrica**: A métrica consiste em quais características foram escolhidas para guiar as decisões de roteamento do algoritmo. Isso é uma questão importante uma vez que a escolha das métricas tem impacto direto no desempenho da rede para atingir o objetivo planejado para o algoritmo.

- **Implementa grafos *uplink*, *Downlink* e Broadcast**: Os três grafos apresentados na especificação *Wireless*HART tem características distintas, logo é esperado que alguns algoritmos lidem de maneira diferenciada com a construção de cada grafo. Este item identifica quais tipos de grafos com os quais a proposta trabalha.
- **Utiliza histórico do nó**: Este item identifica se o algoritmo utiliza os estados passados dos dispositivos (níveis de energia e taxa de sucesso, por exemplo) para se tomar as decisões de roteamento.
- **Implementação e validação**: Diferentes propostas variam em como elas são passadas aos leitores, indo de descrições de políticas até descrevendo passos em um algoritmo de fato. Este item descreve como os autores apresentam a sua proposta e também quais meios de validação utilizados (por exemplo simulações e *testbeds*).

Tabela 3.1: Comparação entre os algoritmos de roteamento para o *Wireless*HART.

Itens/algoritmos	**Alg. Roteamento Han [Han et al. 2011*b*]**	**JRMLR** [Zhang, Yan & Ma 2013]	**Re-add** [Zhang et al. 2014]
Objetivo	Maximizar tempo de vida da rede	Maximizar tempo de vida da rede	Robustez e Maximizar tempo de vida da rede
Métrica	Número médio de saltos (Grau)	Carga de dados no nó, custo de energia para transmissão, energia residual	qualidade do *link*, energia residual e para transmissão, grau
Implementa grafos Uplink, Downlink e Broadcast	Sim	Constrói o *Uplink* e a afirma que o *Downlink* seria construído passando-se o nó de destino, não o *gateway*, como parâmetro	Apenas *Uplink*
Utiliza histórico do nó	Não	Não	Sim
Implementação e validação	Apresenta o pseudocódigo do algoritmo. A simulação foi implementada em [Zand et al. 2012]	Não é apresentado algoritmo, mas resultados de uma simulação Matlab®	Não é apresentado pseudocódigo. Apresenta resultados de simulação mas não são apresentados detalhes da plataforma de simulação.

Objetivo: Em termos de objetivos, todas as propostas apresentaram a mesma preocupação em maximizar o tempo de vida da rede, o que é relevante para dispositivos mantidos por baterias nas RISSF. Entretanto, a proposta mais recente Re-Add [Zhang et al. 2014] inclui em seus objetivos a robustez da rede, reforçando essa característica da tecnologia *Wireless*HART.

Métrica: A comparação apresentada na Tabela 3.1 mostra a procupação das propostas em balancear métricas de modo a balancear os resultados obtidos. Além disso, as propostas JRMLR e Re-Add apresentam o uso de pesos para ajustar a função de custo e definir prioridades para características específicas.

Implementa grafos Uplink, Downlink e Broadcast: Das propostas apresentadas, apenas o algoritmo de roteamento Han apresenta um tratamento específico para cada um dos superframes, significando que é a proposta dentre as analisadas mais próxima da implementação real de um *Network Manager* que possa atender a norma *Wireless*HART.

Utiliza histórico do nó: Como a RISSF é um ambiente dinâmico, ele está sujeito a variações nas comunicações realizadas devido a questões como a interferência, por exemplo. Estas variações podem implicar em uma pequena perda de bits até a incapacitação de canais de transmissão inteiros. Tal situação é relevante para o algoritmo de roteamento de modo que o mesmo possa produzir uma resposta mais adequada a situação de interferência. O uso do histórico do nó pela proposta Re-Add pode ser vista como uma maneira de se evitar que pequenas interferências causem alterações no roteamento e assim propaguem reconfigurações pela rede, causando um aumento no consumo de recursos.

Implementação e validação: Das propostas analisadas, todas utilizam simulações para validação. Mas o algoritmo de roteamento Han tem uma simulação mais consistente devido a sua implementação por um grupo diferente em [Zand et al. 2012], e também pela apresentação de um algoritmo de fato (pseudo-código) para a comunidade.

3.2 Algoritmos de escalonamento de mensagens *Wireless*HART

A literatura apresenta várias abordagens para o escalonamento em redes de sensores sem fio. Por exemplo, podemos citar alguns algoritmos como o apresentado em [Ergen & Varaiya 2010](o qual é baseado em uma técnica de coloração distribuída dos nós), algoritmo distribuído orientado a qualidade de serviço D-SAR em [Zand et al. 2011], e o algoritmo de resposta rápida e consciente de origem SAS-TDMA em [Shen et al. 2013]. Além disso, soluções para minimização de *delay* e *jitter* são apresentadas em [Djukic & Valaee 2009] e [Yu et al. 2013], respectivamente. [Kumar & Chauhan 2011] apresenta uma revisão sobre protocolos MAC baseados em TDMA para RISSF, incluindo orientações para propostas de escalonamento. Embora esta literatura seja relevante, ela não foi especificamente desenvolvida para a tecnologia *Wireless*HART e seus requisitos.

Como esta tese tem como foco a tecnologia *Wireless*HART, esta seção tratará nas principais propostas de algoritmos de escalonamento para esse padrão.

3.2.1 O algoritmo Dang

O trabalho de [Dang et al. 2013] apresenta um algoritmo de escalonamento para a técnica de roteamento por grafo, escalonando recursos de comunicação fim-a-fim com o

objetivo de atender à especificação *Wireless*HART. O algoritmo lida com múltiplas taxas de publicação, logo lidando com múltiplos superframes e buscando ordenar o escalonamento dos *links* de acordo com a sequência de nós indicadas nas tabelas de roteamento.

O algoritmo escalona cada superframe, do de menor tamanho para o de maior tamanho. É importante notar que os dispositivos atribuídos a um dado superframe têm a mesma taxa de publicação, e essa taxa de publicação em comum define o tamanho deste superframe em específico. Uma tabela é utilizada para armazenar o escalonamento, chamada *SlotNumber*, e relaciona o *slot* de tempo com canal utilizado, evitando conflitos de *slot* de tempo e de canal entre as transmissões.

Para escalonar a comunicação entre todos os dispositivos que compõem o caminho entre a origem e o destino, é utilizada uma variável chamada *LastSlot*. Há uma variável *LastSlot* para cada dispositivo e ela armazena o último *slot* de tempo no qual o dispositivo foi escalonado. Quando a próxima transmissão estiver para ser escalonada, ela irá ser escalonada no próximo *slot* disponível após o *LastSlot* (técnica de *First Fit*).

O algoritmo utiliza o esquema de retransmissão baseado no esquema proposto pela especificação *Wireless*HART descrita no Capítulo 2. O algoritmo utiliza duas variáveis para controlar se um *link* que está sendo escalonado é uma retransmissão: a variável *First*, a qual indica se uma transmissão é inicial ou uma retransmissão, e a variável *FirstChannel*, a qual armazena qual canal foi utilizado para transmissão da primeira tentativa, evitando (se possível) o reuso do mesmo canal.

Para validação do algoritmo de escalonamento proposto, o trabalho realiza uma análise do esquema proposto e compara com esquemas de redundância simples, dupla e tripla. Infelizmente não foram apresentados resultados de testes em *testbeds* ou simulação.

3.2.2 O algoritmo Zhang

[Zhang, Zhang, Yan, Xiang & Ma 2013] propõe um algoritmo de escalonamento que utiliza coloração de grafos apenas para o roteamento por grafo e tem por objetivo otimizar a latência e fornecer alta confiabilidade. As arestas representam os *links* da rede e a sua cor (um número inteiro) representa o *slot* de tempo no qual o *link* deve ser escalonado. Duas arestas adjacentes não devem ter a mesma cor. Além disso, um número de canal é utilizado para diferenciar *links* de mesma cor (logo estando no mesmo *slot* de tempo). Todas as arestas na rede devem ser coloridas com pelo menos uma cor.

Para prover confiabilidade, o algoritmo utiliza o ORMGR descrito em [Gao et al. 2013] para determinar uma caminho de transmissão ótimo e utiliza uma política de retransmissão. Esta política considera duas oportunidade de retransmissão dos dados pela

rota ótima. Arestas na rota ótima têm prioridade para serem coloridas e tem duas cores e dois canais diferentes. Com isso, se a transmissão falha por um *link* na melhor rota pela primeira vez, então o dispositivo envia a informação através de outro canal para o mesmo vizinho. A segunda retransmissão ocorre por meio de outro *link*. Entretanto, estas rotas alternativas não têm o *slot* adicional reservado para evitar que se afete negativamente a vazão de dados na rede.

No algoritmo, o grafo é atravessado de origem até o destino colorindo as arestas com base no grau do dispositivo de origem *d*. Quando o *link* relacionado aos nós de grau *d* são escalonados, *d* é decrementado e o algoritmo procede até que o último grau (0, do *gateway*) seja atingido. O algoritmo constrói dois grupos de dispositivos *E* e *E'*. *E* é o grupo de todos os *links* da rede e *E'* é o conjunto dos dispositivos que podem ser coloridos em uma iteração. O grupo *E'* é formado por *links* que conectam nós do grau *d* e os nós da camada superior ou inferior. Um nó pode aparecer no máximo uma vez em *E'* e *links* coloridos da rota ótima devem ser incluídos. Em cada iteração, se há uma aresta não colorida *e* do caminho ótimo, a variável *d* é incrementada e o algoritmo prossegue para o próximo grau. As arestas são coloridas ao final de cada iteração e os canais disponíveis são distribuídos para os *links* alocados. As cores são determinadas pela variável *s*, o qual é incrementada ao fim de cada iteração.

No trabalho, existem testes para propósitos de validação e uma comparação com o algoritmo apresentado em [Dang et al. 2013], mas não foram mostrados maiores detalhes sobre os testes. O trabalho afirma que o algoritmo obteve uma melhor taxa de sucesso que o algoritmo proposto em [Dang et al. 2013] e a alternativa sem retransmissão. Esta vantagem cresce com o aumento da profundidade da rede.

3.2.3 Algoritmo Conflict-aware least laxity first

[Saifullah et al. 2010] apresenta a prova de que o problema de escalonamento na tecnologia *Wireless*HART é NP-Árduo (NP-Hard) e apresenta dois algoritmos para resolver este problema. O primeiro algoritmo proposto é um algoritmo de escalonamento ótimo baseado na técnica de *Branch and bound*. O algoritmo utiliza uma árvore de busca para garantir que o escalonamento será encontrado caso exista. A métrica do algoritmo é o relaxamento do pacote, o qual consiste nos *slots* de tempo restantes menos o número de transmissões que restam para se alcançar o fim da rota.

Embora o algoritmo seja ótimo, o tempo de execução necessário limita a aplicação deste algoritmo em uma rede *Wireless*HART real devido a natureza dinâmica de tal ambiente. Para que seja atendido este requisito, os autores propuseram um segundo algoritmo

baseado na ideia de que o algoritmo deve ter conhecimento dos conflitos entre transmissões, o algoritmo heurístico C-LLF (do inglês, *Conflict-aware Least Laxity First*) baseado na algoritmo LLT (Least Laxity First), utilizado em redes cabeadas. A métrica do relaxamento ciente de conflito consiste em considerar o comprimento das janelas de tempo nas quais as transmissões podem ser escalonadas a a potencial janelas de transmissões concorrentes que causariam conflitos.

As janelas são determinadas pelo valor do tempo de liberação (tempo de *release*, r) de uma transmissão τ_k, sendo o primeiro *slot* no superframe no qual a transmissão pode ser escalonada, e o prazo de entrega ao destino final (*deadline*, d) do pacote. O tempo de *release* e o *deadline* são respectivamente afetados pelo número de saltos até o *link* de transmissão (pre_k) e o número de saltos restantes do *link* ao destinatário ($post_k$). Por exemplo: se a transmissão τ_k dista três saltos do destino, o *deadline* para que esta transmissão ocorra é o *deadline* da entrega do pacote ao destino final menos 3 *slots*.

O algoritmo de escalonamento inicia a partir de um conjunto de todas as transmissões Γ a serem escalonadas e começa alocando desde o primeiro *slot* de tempo até o último (a variável s vai de 0 até o tamanho do superframe). O algoritmo prossegue para escalonar todas as transmissões (τ_k), selecionando um grupo (Released(s)) de transmissões que tem o tempo de *release* igual a s (o *slot* que está sendo alocado no momento). A partir do grupo Released(s), o algoritmo dá prioridade aquelas transmissões com o *deadline* mais próximo e com o maior número de conflitos identificados. Quando uma transmissão é escalonada, todas as transmissões que conflitam são removidas de Released(s) para serem escalonadas em outro *slot*. O algoritmo tenta alocar o máximo de transmissões de Released(s) nos canais disponíveis do *slot* de tempo antes de prosseguir ao *slot* de tempo seguinte. O algoritmo finaliza com sucesso caso ele consiga escalonar todas as transmissões de Γ, mas termina em falha se caso uma transmissão perca seu *deadline* e seja impossível de se escalonar.

Para a validação do algoritmo, o trabalho descreve o uso de simulações e *testbeds* comparando a proposta a um conjunto de políticas de escalonamento. Três métricas foram utilizadas:

- Taxa de Escalonabilidade: Porcentagem de casos de teste para os quais o algoritmo encontrou um escalonamento viável.
- Tamanho do buffer: Máximo de pacotes enfileirados no buffer de um nó quando as transmissões são escalonadas.
- Tempo de execução: tempo de execução médio para a geração do escalonamento para os pacotes criados em um superframe.

A proposta tem melhor desempenho quando comparados com as outras políticas na maioria dos testes, a exceção do tempo de execução, o qual cresceu mais que o dos concorrentes com o aumento do número de nós da rede. Contudo, o tempo de cálculo obtido foi da ordem de 2,5 segundos, o qual é um tempo de computação razoável para uma rede *Wireless*HART.

3.2.4 O Algoritmo de escalonamento Han

O algoritmo apresentado em [Han et al. 2011*b*] utiliza a política *Fastest Sample Rate First* (FSRF) que dá a prioridade de escalonamento para os dispositivos com taxa de amostragem (semelhante a taxa de publicação referida anteriormente) mais rápida. O escalonamento é baseado nos grafos confiáveis apresentados na seção 3.1.1 e utilizam uma função recursiva para escalonar os *links* na forma de uma matriz global $\mathcal{M}$. Esta função recursiva aloca os nós da origem ao destino utilizando um algoritmo de busca em profundidade. A matriz $\mathcal{M}$ representa todas as alocações de canal/*slot* de tempo para cada *link*; Suas dimensões são dadas pelo número de canais (16) pelo tamanho do superframe mais longo. O superframe $\mathcal{F}_i$ representa um grupo de dispositivos com a taxa de amostragem r_i. Uma variável l_i representa o tamanho do superframe $\mathcal{F}_i$.

Geralmente, cada dispositivo está a vários saltos do seu destino. Se o *Network Manager* aloca todos os recursos dos múltiplos caminhos no escalonamento, a escalonabilidade da rede diminui rapidamente. Para evitar essa situação, o algoritmo utiliza um superframe paralelo $\mathcal{F}_i'$, com tamanho igual a $l_{\mathcal{F}'} = 2 * l_{\mathcal{F}}$, para acomodar estas transmissões de nós com duas possíveis rotas (ver Figura 3.4 adaptada de [Dickow 2014]). Isso é feito reduzindo a taxa de amostragem pela metade e determinando um *offset* que posicione o *link* em um padrão de comunicação idêntico ao original mas por diferentes rotas. É importante notar que reduzindo a taxa de amostragem, por exemplo, um dispositivo que publica uma vez por segundo será equivalente a um dispositivo que publica dados uma vez a cada a cada dois segundos.

A alocação é dividida pelo tipo do *link* a ser escalonado. O *link* pode ser de *uplink*, *downlink* ou compartilhado. Um *link* exclusivo é um ocupado por dois dispositivos de comunicação apenas (transmissor e receptor). *Links* compartilhados permitem que múltiplos dispositivos disputem a transmissão para um outro dispositivo simultaneamente. A divisão pode ser vista na Tabela 3.2.

Links compartilhados podem ser utilizados por dispositivos para retransmissões, assim melhorando a confiabilidade da rede. Se uma tentativa de escalonamento falha, o *Network Manager* envia uma mensagem de falta de recursos de banda para o nó solicitante.

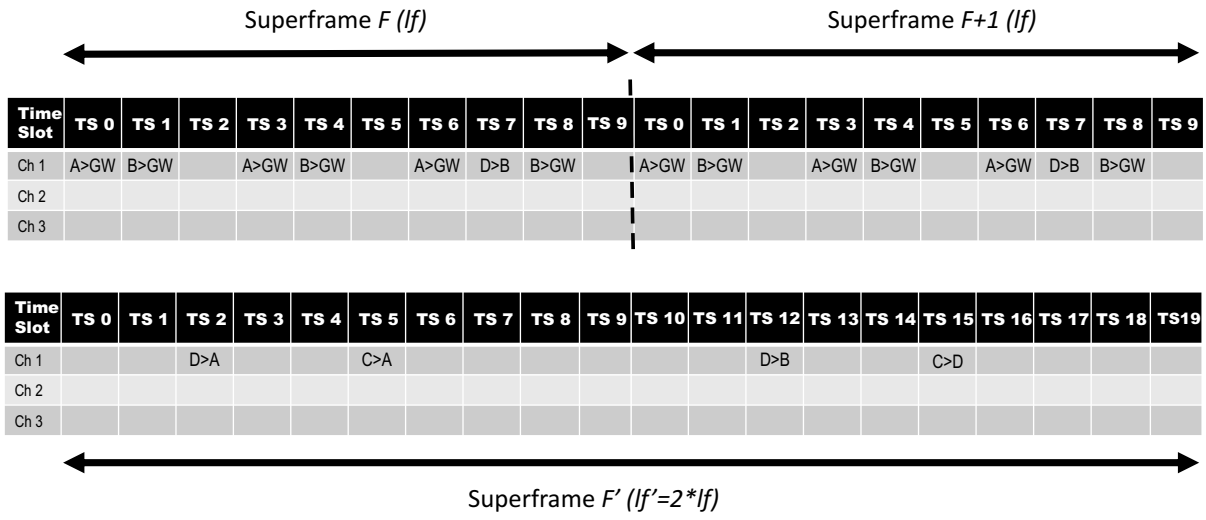


Figura 3.4: Superframes $\mathcal{F}'$ e $\mathcal{F}$.

Tabela 3.2: Divisão do superframe para o algoritmo de escalonamento.

Tipo do Link	**Primeiro Slot de tempo Possível**
Uplink/Slot exclusivo	0
Uplink/Slot Compartilhado	$\frac{l_i}{4}$
Downlink/Slot exclusivo	$\frac{l_i}{2}$
Downlink/Slot Compartilhado	$\frac{3*l_i}{4}$

A implementação do escalonamento foi validada utilizando-se simulação e apresentou uma melhora, principalmente, na métrica de utilização da rede enquanto mantinha uma taxa de sucesso de escalonamento próximo a 100% para taxas de amostragem mais baixas.

É importante destacar que o algoritmo de escalonamento Han assume uma rede que possua pelo menos dois *access points* ligados ao *gateway* de modo melhorar a confiabilidade da rede.

3.2.5 A política Zhang

[Zhang et al. 2011] estabelece um limite inferior de tempo de *convergecast*, estabelecendo um limite inferior para o número de canais necessários para um *convergecast* em tempo ótimo e propõe duas políticas de escalonamento baseadas no escalonamento ótimo

para topologias em linha proposto por [Zhang et al. 2009]. É importante notar que no trabalho não foram foram apresentados algoritmos e nenhuma técnica de redundância foi adotada. As políticas propostas se aplicam a topologia de múltiplas linhas (*Multi-line topology*) como a apresentada na Figura 3.5. É importante notar que a questão principal abordada por esta proposta é o fato de que o último salto para o *gateway* é uma limitação a todo o sistema uma vez que todos os pacotes devem passar por este salto. De tal modo, um único rádio do *gateway* permite que apenas um pacote seja transmitido por *slot* de tempo.

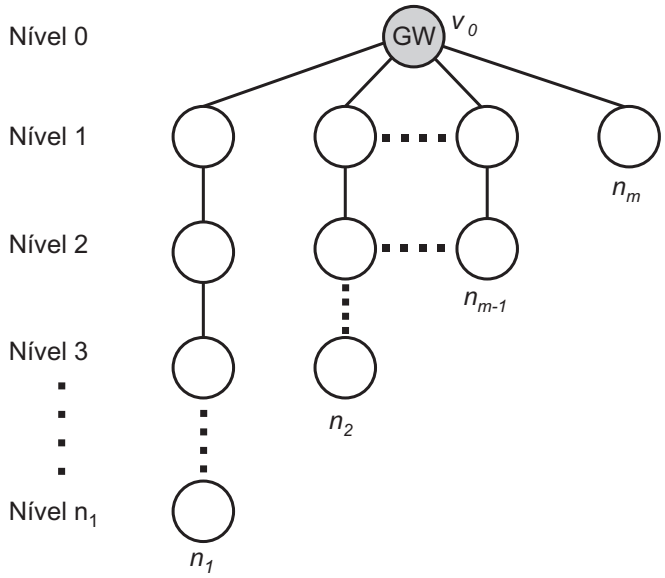


Figura 3.5: Exemplo de topologia de múltiplas linhas.

A primeira política garante um tempo ótimo de *convergecast*, mantendo o limite inferior estabelecido para o tamanho do superframe necessário para alocar todas as transmissões. Entretanto, esta política não é eficiente em termos de utilização de canais. Para atingir algo próximo ao ótimo neste sentido, a segunda política proposta utiliza o limite inferior estabelecido para o número de canais, apesar de que este limite não seja sempre possível de ser alcançado. A segunda política (chamada no original *time- and channel-optimal convergecast policy*) é uma abordagem heurística que se baseia em dois passos que devem ser aplicados a cada *slot* de tempo.

O primeiro passo é chamado "manutenção da conectividade"(*connectivity keeping*), atravessa os níveis da topologia do *gateway* ao nós folhas e prioriza o escalonamento de *links* que evitem os chamados buracos no escalonamento (*schedule holes*). Os buracos no escalonamento são aqueles momentos nos quais o *gateway* não recebe um pacote em um

slot de tempo mesmo que a rede ainda tenha pacotes destinados ao mesmo. Em outras palavras, a oportunidade de transmissão para o *gateway* são subutilizadas. Tal situação provoca atrasos no escalonamento da rede.

Se o número de dispositivos escalonado na manutenção da conectividade não atinge o máximo da capacidade de um *slot*, a política chamada "escalonamento baseado em *deadline*"(*deadline-based scheduling*) deve ser aplicada. Esta política organiza os dispositivos que podem ser escalonados para a transmissão nesse *slot* de tempo e as ordena em ordem não-decrescente, dando prioridade para aqueles com *deadlines* mais estritos. O número de saltos é utilizado como um critério de ordenação secundário.

Um algoritmo heurístico baseado na segunda política foi implementado em uma simulação na plataforma Matlab® e em um simulador COOJA [Osterlind et al. 2006]. Para a simulação em Matlab®, a proposta obteve um escalonamento que utilizou no máximo um *slot* a mais que o limite inferior estabelecido para o tamanho do superframe. Das 10.000 execuções, apenas duas não alcançaram o limite inferior para o tempo de *convergecast*. O teste no simulador COOJA mostrou que o esquema consegue coletar dados de 34 dispositivos divididos em seis linhas (as linhas tem respectivamente 10, 8, 6, 4, 4, e 2 dispositivos) e 0,4 segundos para uma carga de 125 bytes.

3.2.6 Algoritmos de escalonamento não *Wireless*HART

Nesta seção, assim como foi feito na seção 3.1.4 para os algoritmos de roteamento, serão apresentados alguns dos mais novos algoritmos de escalonamento para RISSF que podem servir como ideias para futuras aplicações na tecnologia *Wireless*HART.

[Zand et al. 2013] apresenta o esquema distribuído de gerenciamento de redes para aplicações em tempo real (D-MSR, do inglês *Distributed Network Management Scheme for Real-Time applications*), um sistema de gerenciamento de redes que tem o objetivo de prover comunicação confiável em tempo real e cumprir com os requisitos de vazão de dados para a aplicações de controle industrial. Para o propósito do escalonamento, o sistema adota o algoritmo D-SAR (Distributed Scheduling algorithms for Real-Time applications) apresentado em [Zand et al. 2011]. D-SAR utiliza um conjunto de mensagens que atravessam a rede da origem ao destino baseadas na rotas da configuração de roteamento. Cada dispositivo inclui parâmetros como a lista de células canal-*slot* de tempo normalmente não utilizados do próximo salto, um endereço de destino, ID do tráfego de dados e a célula de canal-*slot* de tempo selecionada no último salto. Com essa informação, os dispositivos organizam suas próprias matrizes de escalonamento e passam a mensagem adiante para o destino. É importante destacar também que o D-MSR utiliza a

técnica de circuito virtual para prover comunicação em tempo real.

[Cheng et al. 2014] Propõe um algoritmo de escalonamento para RISSF orientado a serviço e baseado em Campos Aleatórios de Markov (Markov Random Field - MRF) chamado *MRF-based Multi-service Node Scheduling* (MMNS). É importante notar que o algoritmo proposto é uma solução heurística. O seu objetivo é gerenciar dispositivos com equipados com múltiplos sensores (por exemplo som, movimento e temperatura) de modo que um mesmo nó pode prover vários serviços se sensoriamento. A proposta pretende organizar a divisão dos recursos entres o vários serviços de uma maneira energeticamente eficiente. O algoritmo MMNS é baseado em dois outros algoritmos desenvolvidos em [Cheng et al. 2014]: o algoritmo *Multi-service Data Denoising* (MDD) e o *Representative node Selection and service Determination* (RSD). O primeiro tem o objetivo de minimizar o nível de ruído nos dados captados. O segundo pretende detectar nós representativos da rede e os serviços providos pelos mesmos.

[Nhon & Kim 2015] difere dos outros trabalhos da área uma vez que este apresenta dois métodos de escalonamento de *slots* compartilhados para redes ISA100.11a. O primeiro, chamado *Traffic-aware message scheduling* (TAMS), utiliza informações sobre o trafego dos dados para definir grupos de dispositivos e decidir em quais ciclos os grupos irão transmitir para que a banda seja utilizada da melhor maneira, evitando-se colisões em cada ciclo. O segundo algoritmo é o *Contention Window Size Adjustment* (CWSA), o qual utiliza as informações do tráfego de dados para ajustar o tamanho da janela de contenção (fixada em 192 microssegundos pela norma do ISA100.11a, independente do tráfego) quando a probabilidade de colisão em um *slot* de tempo excede um valor limite.

O algoritmo desenvolvido por [Yu et al. 2013] tem por objetivo a redução do *jitter* provocado principalmente por múltiplas chegadas de cada pacote por rotas diferentes e pela deriva entre as posições dos *slots* iniciais e finais em um superframe. Para realizar este objetivo, o algoritmo classifica as transmissões em tarefas que as agrupam de modo a levar os dados da origem ao destino. Após as definições das tarefas de modo a evitar-se conflitos, a escalonabilidade é garantida dando-se prioridade às tarefas baseadas no *deadline* para aquela tarefa e no número de saltos necessários.

[Djukic & Valaee 2009] descreve um método que encontra um escalonamento TDMA sem conflitos e minimiza o atraso em uma rede TDMA *multi-hop* sem fio. A solução é dividida em duas partes. A primeira consiste em encontrar uma ordem de transmissão que atenda os requisitos de precedência para a comunicação. A solução utiliza programação inteira de modo a encontar a ordem das transmissões que minimizem o atraso de escalonamento máximo. A segunda parte consiste em construir a ordem das transmissões de acordo com um grafo de conflitos ([Jain et al. 2003], [Ramanathan 1999]) para o algo-

ritmo de escalonamento (o utilizado como exemplo foi o de Bellman-Ford) de modo que as transmissões sejam acomodadas no escalonamento sem conflitos.

O trabalho em [Shen et al. 2014] apresenta um novo protocolo de acesso ao meio baseado no *Wireless*HART e no ISA100.11a que foca na resolução das questões do tráfego de dados críticos e de latência das RISSF. Como característica principal, ele utiliza as partes iniciais e finais dos *slots* de tempo tradicionais de modo que dispositivos com prioridade maior possam sinalizar para dispositivos de menor prioridade que uma transmissão de maior prioridade irá ocorrer, assim os dispositivos devem reter transmissões de baixa prioridade que estariam escalonadas para aquele *slot* de tempo. Este esquema permite pacotes com maior prioridade utilizem a banda de transmissões de menor prioridade durante o funcionamento da rede. Entretanto, essa técnica não é aplicável na versão atual da especificação *Wireless*HART, uma vez que altera a estrutura dos trâmites de sinais dentro do *slot* de tempo.

O trabalho apresentado em [Zhang et al. 2012] combina o TDMA tradicional com o *Sloted Aloha* de modo a produzir um escalonamento ótimo para uma rede de sensores de salto único com restrição de atraso. Este algoritmo tem por objetivo melhorar a confiabilidade enquanto evita o desperdício de recursos da rede, uma situação comum em redes TDMA. O algoritmo aperfeiçoa os trabalhos apresentados em [Ahn et al. 2006] e [Rhee et al. 2008] utilizando o Sloted Aloha ao invés do CSMA, evitando a desvantagem do atraso variável provocado pelo *backoff* exponencial do CSMA. Dois algoritmos são apresentados: um que calcula um escalonamento ótimo e um algoritmo heurístico que calcula soluções próximas às soluções ótimas mas com um custo computacional mais modesto. Em resumo, a estratégia consiste no seguinte: baseado em um modelo de perdas de Bernoulli, cada transmissão tem uma probabilidade de sucesso p_i, e a partir dela o algoritmo distribui as transmissões dentro do escalonamento de modo a maximizar a probabilidade de sucesso do mesmo. Cada *slot* de tempo pode ser um *slot dedicado*, onde apenas uma transmissão é escalonada para ocorrer, ou um *slot compartilhado*. Em um *slot* compartilhado, os dispositivos escalonados podem tentar uma transmissão utilizando o esquema Slotted Aloha caso as suas tentativas nos *slots* dedicados tenham falhado. A solução ótima consiste em tentar todas as combinações possíveis de modo a maximizar a probabilidade de sucesso do escalonamento. A solução heurística consiste em um algoritmo guloso que elege o(s) próximo(s) dispositivo(s) a serem escalonados de modo a maximizar a probabilidade do controlador receber mais de um pacote no passo $k+1$ com base no k-ésimo passo anterior. Os resultados das simulações numéricas em [Zhang et al. 2012] mostram que o esquema supera a abordagem TDMA tradicional de *slots* exclusivos, fazendo do uso de *slots* compartilhados um tópico relevante para futuros trabalhos.

O trabalho [Akerberg, Gidlund, Lennval, Jonas & Bjorkman 2011] foca na integração da segurança dos dados e da estrutura das RISSF e propõe um *framework* para prover comunicações confidenciais e seguras para a rede. Como o foco do nosso trabalho é o escalonamento e o roteamento, é importante notar que o trabalho em [Akerberg, Gidlund, Lennval, Jonas & Bjorkman 2011] expressa preocupação sobre o foco das atuais implementações do *Wireless*HART na transmissões de *uplink* em detrimento das tranmissões de *downlink* e como isso afeta a segurança das redes de controles distribuídos baseados em *Wireless*HART (*WirelessHART-based Distributed Control Systems* - DCS). Por exemplo, o atraso na entrega de um sinal de desligamento enviado do controlador para o atuador em um estado crítico pode provocar questões de segurança. Os autores em [Akerberg, Gidlund, Lennval, Jonas & Bjorkman 2011] afirmam que para que se atinjam bom resultados todos os sinais de referência devem chegar aos atuadores no mesmo ciclo, e o *jitter* e o *delay* devem ser reduzidos a um mínimo. Para isso, é sugerido a implementação de um novo comando *Wireless*HART para que a aplicação de controle configures transmissões periódicas aos atuadores (transmissões de *downlink*).

O trabalho apresentado em [Yan et al. 2014] tem por objetivo maximizar a confiabilidade de comunicações fim-a-fim em RISSF. O trabalho organiza a topologia de rede em um conjunto de hipernós. Os hipernós consistem em um conjunto de dispositivos que se conectam diretamente entre si, formando um grafo completo. A organização lógica de um conjunto de hipernós forma um hipergrafo. Finalmente, em termos de escalonamento, o trabalho propõe a divisão do escalonamento em duas partes. A primeira é a dedicada ao escalonamento, a qual decide a quantidade de *slots* que devem ser alocadas para cada hipernó de modo a encaixar as transmissões periódicas de pacotes entre os dispositivos de origem e destino. O segundo esquema de escalonamento, chamado "esquema de *slot* compartilhados"(*shared slot scheme*) permite aos hipernós utilizar os *slots* de tempo restantes de transmissões bem sucedidas para melhorar a flexibilidade e a confiabilidade das comunicações. É importante notar que o esquema proposto dá suporte aos roteamentos *sigle-path* e *any path* introduzidos em [Biswas & Morris 2005].

Apresar do trabalho apresentado em [Saifullah et al. 2011] não ser um algoritmo de escalonamento, ele lida com a questão de atribuição de prioridades aos fluxos de dados em redes *Wireless*HART. Esta questão consiste em atribuir prioridades ao fluxos de dados em tempo real, de modo que estes fluxos atinjam seu *deadline* e também sirvam para trabalhar em conjunto com testes de escalonabilidade, de modo que a rede possa realizar um planejamento efetivo de capacidade de rede, um controle de admissão eficiente e adaptações de topologia. Como a atribuição de prioridade é um problema NP-Árduo, o trabalho primeiro propõe um algoritmo de atribuição de prioridades ótimo baseado em

busca local para qualquer caso de analise de *delay* dado. Entretanto, em se tratando de eficiência, o trabalho propõe um algoritmo de busca heurística para a atribuição de prioridades.

O trabalho em [Li et al. 2014] lida com o problema de redes de salto-simples (*single-hop*) com tráfegos heterogêneos e restrições de *delay* de pacotes. O objetivo do trabalho é fornecer uma alocação ótima de *slots* de tempo e escalonar a transmissão dos pacotes. O autor divide a solução para este problema em duas partes. A primeira parte é chamada de "problema da alocação de *slots* em sub-períodos"(*subperiod slot allocation problem*), o qual consiste em definir quantos *slots* cada dispositivo terá disponível para realizar transmissões/retransmissões com base na probabilidade de sucesso de cada *link*. Esta parte do problema é NP-Árdua e foi modelada como um problema de programção inteira não linear. Desse modo, os autores propõem a transformação do problema em um problema de programação linear inteira e então apresentam um algoritmo que calcula a solução ótima em tempo polinomial para esta versão simplificada do problema.

A segunda parte do problema consiste em decidir quais *slots* serão alocados para cada dispositivo de modo a atender a restrição da taxa de publicação de cada um deles e como estas transmissões alocadas se relacionam com as transmissões de outros dispositivos. Isto é cumprido alocando-se o número de *slots* definidos para cada dispositivo em um sub-período de tempo definido taxa de atualização de cada dispositivo e então combinando estes sub-períodos em um hiperperíodo. O hiperperíodo tem a duração de mínimo multiplo comum das taxas de atualização apresentadas na rede de modo a acomodar vários sub-períodos sem que haja fracionamento nos mesmos. Entretanto, a técnica não é aplicável ao atual *Wireless*HART devido ao seu foco em redes sem fio de único salto, diferindo da estrutura de múltiplos saltos. Contudo, a modelagem do sistema proposto é compatível com o a especificação *Wireless*HART e pode ser aplicada nas estruturas de cluster (com salto único) das tecnologias WIA-PA e ISA100.11a.

3.2.7 Comparação dos protocolos de escalonamento

Para comparar as propostas de algoritmos de escalonamento previamente apresentados para a tecnologia *Wireless*HART, apresentamos nesta seção um conjunto de itens a serem comparados entre os algoritmos. Estes itens são:

- **Objetivo**: Desenvolvedores de um algoritmo podem possuir diferentes aplicações, situações e necessidades as quais necessitam soluções igualmente diferentes. Isso nos leva ao desenvolvimento de uma variedade de algoritmos diferentes e que, por sua vez, possuem objetivos diferentes. Então definimos aqui objetivo como sendo

aquilo que a proposta de algoritmo pretende melhorar quando utilizado em uma rede.

- **Métrica**: O algoritmo necessita de parâmetros para realizar as escolhas que constroem o escalonamento e priorizam os objetivos definidos pelo desenvolvedor dos algoritmos. As métricas são aqueles parâmetros (por exemplo: grau do nó e *deadline* de transmissão) que são utilizados para determinar os canais e *links* que serão escalonados.
- **Múltiplos superframes**: A especificação *Wireless*HART define que múltiplas taxas de publicação devem ser mantidas de modo a lidar com diferentes tipos de dispositivos e processos. Dispositivos com a mesma taxa de publicação são agrupados em um mesmo superframe. O item *Múltiplos superframes* identifica se a proposta afirma explicitamente como lida com a questão dos superframes.
- **Redundancia**: As propostas podem enxergar o problema do escalonamento de duas maneiras: uma é ver o problema como um conjunto de transmissões que devem ser escalonadas; a outra é ver um grupo de transmissões fim-a-fim que devem ter todos os saltos que compõem essa comunicação escalonados e providos com redundância. O item *Redundância* descreve como a proposta utiliza ou sugere um método de redundância.
- **Implementação**: Diferentes métodos podem ser utilizados para validar uma proposta. Isso pode variar de abordagens mais simples, como a análise da solução, até a implementação de um *testbed* completo com dispositivos reais. Há ainda casos nos quais a proposta apresenta o código de fato da solução, mas há também casos em que apenas uma política ser utilizada é apresentada. Para elucidar tais questões, o item *Implementação* mostra se o trabalho analisado apresenta o pseudocódigo para o algoritmo e quais meios de validação foram utilizados.
- **Diferenciação de fluxo**: A especificação *Wireless*HART classifica o fluxo de informações para ida e vinda do *gateway* (*sink*, *source*) como *Uplink* e *Downlink*, respectivamente. Como estes fluxo têm características diferentes, o item *Diferenciação de fluxo* identifica como o trabalho lida especificamente com os fluxos de *downlink* e *uplink* da rede *Wireless*HART.

A Tabela 3.3 apresenta as características de cada proposta e tem a intenção de identificar quais tendências estão prevalecendo para o escalonamento no *Wireless*HART, dando um direcionamento para futuros melhoramentos na área.

Objetivo: É possível observar que existem dois principais tópicos para o objetivo. O primeiro inclui aqueles cuja preocupação principal é preencher o escalonamento, como

Tabela 3.3: Comparação de algoritmos de escalonamento *Wireless*HART.

Característica / Proposta	**Dang Alg.** [Dang et al. 2013]	**Zhang Alg.** [Zhang, Zhang, Yan, Xiang & Ma 2013]	**C-LLF**[Saifullah et al. 2010]	**Han Alg.**[Han et al. 2011*b*]	**Zhang Policy**[Zhang et al. 2011]
Objetivo	Acomodar todas as TX	Latência e alta confiabilidade	Encontrar o escalonamento se ele for factível	Confiabilidade	Utilização de tempo e canal
Métrica	Taxa de publicação	Grau do nó	Janela de tempo (Relaxação do nó)	Taxa de publicação	Deadline e Contagem de saltos
Múltiplos superframes	Sim	Não	Não	Sim	Não
Redundância	Sim	Sim	Não	Sim, Usa slots compartilhados para retransmissões	Não
Implementação	Sem algoritimo. Usa validação por comparação analítica.	Sim, *testbed* mas sem detalhes sobre a implementação	Sim, algoritmos ótimo e heurístico; Validação com *testbeds* e simulação	Sim, com simulação em [Han et al. 2011*b*] e simulação completa em [Zand et al. 2012]	Nenhum algoritmo é apresentado mas utiliza simulações em Matlab® e COOJA para validação
Diferenciação de fluxo	Não	Não	Não	Sim	Não

o algoritmo Dang [Dang et al. 2013] e o C-LLF, já que este problema já é naturalmente complexo. A segunda tendência é formada por algoritmos que são propostos especificamente para o melhoramento de características específicas como latência, confiabilidade, uso de tempo e canal.

Métrica: um equilíbrio de utilização entre as métricas grau (no caso o mesmo que contagem de saltos), *deadline* (relaxação, envolve o *deadline* para ser calculado) e taxa de atualização, uma vez que podemos observar que todas propostas utilizam uma ou mais delas. Isso mostra uma clara tendência para o uso dessas métricas, mas deixa espaço para novas, como a robustez.

Múltiplos Superframes: A preocupação com múltiplos superframes é apenas explícita nas propotas do Algoritmo Dang e do Algoritmo Han, as quais utilizam a taxa de publicação como métrica para decisões. As outras propostas não especificam como lidam com a questão e deixam isso para ser resolvido pela implementação do *Network Manager*.

Redundância: Três propostas apresentaram soluções com preocupações sobre redundância, se mostrando soluções mais completas. A solução apresentada pelo algoritmo Han [Han et al. 2011*b*] tem uma característica em particular: a proposta utiliza os *slots* compartilhados para possíveis retransmissões; logo, os dispositivos que necessitam de retransmissão competem pela possibilidade de utilizar o meio de transmissão.

Implementação: a maioria das propostas implementam suas ideias utilizando simulações, o que é esperado devido a questões de custo e confiabilidade, como pode ser visto em [Nobre et al. 2010]. Os algoritmos C-LLF e Zhang utilizam *testbeds*, mas o segundo não provê detalhes de como isso foi feito. A forma de validação mais destoante apresentada foi a comparação analítica utilizada pelo algoritmo Dang, se baseando no número de saltos e confiabilidade dos *links* para determinar o melhor desempenho da técnica.

Diferenciação de fluxos: Apenas o algoritmo Han [Han et al. 2011*b*] implementa a diferenciação de fluxo de modo a definir que região do superframe seria ocupada pro cada fluxo, como apresentado na tabela 3.2. Todas as outras propostas lidam com os fluxos de maneira indiscriminada.

3.3 Simuladores *Wireless*HART

Quando se desenvolve um novo protocolo ou algoritmo para RISSF, é muito importante medir o desempenho da proposta de modo a determinar os cenários mais adequados para sua utilização, seus pontos fortes e fracos. Também é importante comprar o desempenho da proposta com aquele de soluções já criadas. Para se fazer isso existem duas possibilidades: a implementação de um *testbed* com dispositivos reais e a simulação computacional. Embora *testbeds* forneçam resultados realistas, os custos e complexidade do ambiente de testes, principalmente nos seus estágios mais iniciais, pode tornar esta opção inviável. Contudo, a simulação computacional pode ser utilizada como uma solução escalável de baixo custo para testes de novas propostas antes de investir recursos em redes reais. Além disso, simulações podem também ser utilizadas como guia para usuários na hora de decidir qual tecnologia RISSF utilizar em sua aplicação. Diante desta situação, apresentaremos nessa seção uma descrição dos principais simuladores *Wireless*HART, sua características e onde os encontrar.

O simulador *COOJA* (*Contiki OS Java*) *Simulator* é um simulador para o Contiki OS. O Contiki OS é um sistema operacional (OS) desenvolvido para funcionar em dispositivos de recursos limitados (por exemplo bateria, memória e processamento) como sensores [Osterlind 2006]. Como o COOJA simula cada nó como dispositivos que roda um sistema operacional de fato, o código utilizado em dispositivos reais pode ser utili-

zado diretamente no simulador. O objetivo principal do COOJA é prover extensibilidade à simulação. Para alcançar isso, o simulador implementa duas estruturas *plugins* e interfaces. *Plugins* são geralmente inicializados antes de serem utilizados e permitem iteração com o simulador. Interfaces interagem com sensores, simulando periféricos de hardware ou monitorando valores de interesse. Para simular o *Wireless*HART, [Konovalov 2010] desenvolveu uma solução que consiste em adicionar um dispositivo intermediário habilitado para o *Wireless*HART operando nas camadas física e de enlace, as quais podem lidar com tarefas de tempo crítico do protocolo. Além disso, de acordo com a política Zhang, [Zhang et al. 2011] afirma ter utilizado o COOJA mas poucos detalhes sobre como o simulador opera foram mostrados. Finalmente é importante notar que o COOJA é um software aberto.

O *Network Simulator 2* (NS-2) é um simulador de eventos discretos de código aberto para pesquisa sobre redes. O primeiro NS começou com uma modificação do *REAL Network Simulator* em 1989. Atualmente, o desenvolvimento do NS é mantido por um esforço colaborativo de diversas instituições e pesquisadores. O simulador provê suporte para simulações TCP, roteamento, e protocolos *multicast* em redes cabeadas ou sem fio, comunicações locais ou por satélite [NS-3 Consortium 2015]. O NS-2 utiliza duas linguagens de programação: C++ é utilizado para codificar os protocolos a serem utilizados e a linguem interpretada *Object oriented Tool Command Language* (OTCL) é utilizada para configurar os parâmetros de simulação. Para um melhor entendimento das simulações, o NS-2 possui o modulo *Network Animator* (NAM), uma ferramenta que anima os registros das simulações completas. A implementação para o NS-2 pode ser encontrada em [Zand et al. 2012].

Embora o NS-2 seja um simulador popular, ele apresenta problemas em relação a escalabilidade, uso de memória e tempo de simulação [Weingärtner et al. 2009]. Além disso, a sua documentação está desatualizada e utiliza grandes abstrações em suas camadas inferiores [NS-3 Consortium 2015]. Para lidar com estas questões, os desenvolvedores criaram o projeto *Network Simulator 3* (NS-3), um sucessor natural do NS-2. O novo simulador utiliza uma estrutura atualizada, modular e orientada a objetos com uma vasta documentação disponível. A estrutura modular foi concebida de maneira que os módulos podem ser implementados para serem reutilizados em implementações de outras tecnologias. O NS-3 trabalha com a linguagem C++ e, opcionalmente, Phyton. Um módulo para o NS-3 com um foco na modelagem das camadas inferiores, com calculo de consumo de energia e probabilidades de erro independentes para cada *link* pode ser encontrado em [Nobre et al. 2010] e [Nobre et al. 2015*a*].

O Matlab® é um programa proprietário para a comunidade científica que fornece fer-

ramentas para um amplo conjunto de campos, de redes neurais a aeronáutica. Em virtude dessa popularidade, a ferramenta naturalmente se tornou uma plataforma de desenvolvimento de módulos de simulação de redes. [Kostadinovic et al. 2009] apresenta uma modificação no *Wireless Network Block* do simulador TrueTime de modo a simular redes *Wireless*HART. As modificações foram feitas utilizando-se funções C++ e interfaces MATLAB MEX, e as simulações são configuradas utilizando a linguagem de blocos do Simulink. É importante notarmos que o algoritmo JRMNL apresentado anteriormente foi validado utilizando Matlab® [Zhang, Yan & Ma 2013], mas não foi desenvolvido um módulo de simulação para a plataforma.

Outra abordagem utilizando o TrueTime foi implementada por [Shah et al. 2010] com o foco nos níveis de aplicação, principalmente laços de controle. O módulo provê informação sobre possíveis erros nas dependências do fluxo de dados e conflitos de acesso na rede. Entretanto, a implementação não dá suporte a comunicação de múltiplos saltos e a utilização dos 15 canais definidos pela especificação *Wireless*HART [(IEC) 2010].

O OPNET é um simulador de redes com eventos discretos proprietário que adota uma estrutura hierárquica dividida em três partes (rede, nó e processo) [Hao et al. 2011]. Uma implementação *Wireless*HART é apresentada em [Gao et al. 2012] e tem por objetivo de construir toda a pilha *Wireless*HART. Este modelo de rede pode ser utilizado como plataforma de testes para algoritmos de escalonamento e roteamento. Outa implementação pode ser vista em [Wang & Barac 2013], onde o autor afirma que a implementação é "Uma implementação primária da camada controle de acesso ao meio *Wireless*HART"e propõe um melhoramento no método de acesso dos slots compartilhados, utilizando CSMA/CA (*Carrier Sense Multiple Access/Collision Avoidance*) ao invés do ALOHA existente. A linguagem utilizada no OPNET é uma linguagem de blocos proprietária com a possibilidade de agendamento de eventos. É importante notar que atualmente o OPNET está disponível como parte do *Riverbed Network Simulation Software* [Riverbead Technology 2015].

Para um melhor entendimento a Tabela 3.4 analisa algumas questões chave sobre os supracitados módulos de simulação *Wireless*HART.

Tabela 3.4: Módulos de simulação *Wireless*HART disponíveis.

Simulador	Open Source	Linguagem	Ref. Módulo	Ref. Simulador
COOJA	Sim	JAVA	[Konovalov 2010]	[Contiki 2015]
NS-2	Sim	OTCL C++	[Zand et al. 2012]	[USC 2015]
NS-3	Sim	C++ Python	[Nobre et al. 2010, Nobre et al. 2015*a*, Nobre et al. 2014]	[NS-3 Consortium 2015]
Matlab® (TrueTime)	Não	Simulink blocks	[Kostadinovic et al. 2009] [Shah et al. 2010]	[Mathworks 2015]
OPNET	Não	OPNET Linguagem de Blocos	[Gao et al. 2012]	[Riverbead Technology 2015]

Capítulo 4

Escalonamento Flow

Neste capítulo será descrita a proposta de escalonamento de mensagens que foi denominada Escalonamento Flow (escalonamento por fluxos), assim como um detalhamento das métricas utilizadas e seu algoritmo de funcionamento. Como mostrado no Capítulo 3, existem na literatura apenas algumas soluções de escalonamento desenvolvidas especificamente para a tecnologia *Wireless*HART. Dentre elas podemos destacar o Escalonamento Han [Han et al. 2011*b*], sendo, inclusive, implementada para simulação em [Zand et al. 2012]. Contudo, o escalonamento Han apresenta algumas limitações. Podemos ressaltar que o algoritmo dá prioridade de escalonamento aos dispositivos que tem as menores taxas de publicação (*Publish Rate* - PR). Dentro de cada conjunto de PR, o algoritmo utiliza uma técnica de *First-fit*, que aloca o *link* na primeira opção possível disponível dentro do *superframe*. Isto é feito em sequência de modo que se preserve a ordem das transmissões para que a rota entre a origem e o destino seja construída conforme descrito na Seção 3.1.1. Em casos como o de um conjunto de sensores por composto de vários dispositivos com PR iguais, o algoritmo se resume a um algoritmo de *First-Fit* sem qualquer outro parâmetro para otimização. Além disso, conforme descrito na Figura 4.1, quando um nó com dois vizinhos (utilizando rotas primária e secundária no *superframe* F') encaminha os pacotes para um nó com um único vizinho (utiliza apenas a rota primária e no *superframe* F), há uma reserva de *slots* que não serão utilizados. Tais *slots* estão exemplificados na Figura 4.1 pelas marcações cinza dos *superframes*. Esta situação gera um dispêndio de *slots* que poderiam ser utilizados para transmissões, além de impedir que *links* que envolvam os dispositivos transmissores e receptores dos *slots* sejam escalonados neste mesmo *slot* de tempo.

Ciente dessa deficiência, neste capítulo iremos apresentar uma proposta de algoritmo de escalonamento de mensagens baseado no conceito de fluxo, que por isso é denominado Flow. No nosso caso, um fluxo consiste no envio de um pacote da origem ao destino em uma rede incluindo todas as transmissões entre os nós na rota. Tendo este conceito

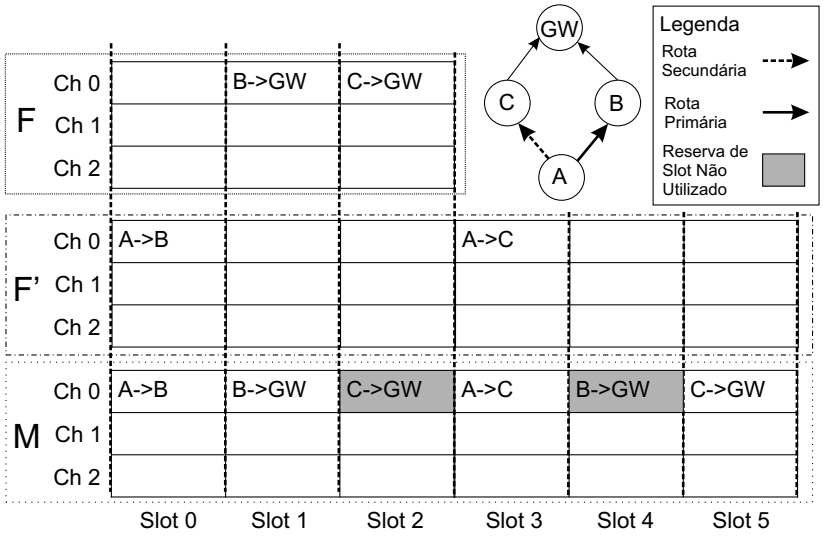


Figura 4.1: Como o slot pode ser desperdiçado no escalonamento Han.

de fluxo em vista, o escalonamento de mensagens é realizado organizando-se todas as transmissões a serem realizadas em um conjunto de fluxos que são escalonados dentro do *superframe*.

4.1 Métricas e critério de decisão

Nesta seção serão detalhadas as métricas (Grau, Janela de Transmissão e Criticalidade) utilizadas no Algoritmo de Escalonamento Flow, indicando como estas métricas são combinadas para ordenação de prioridades dos fluxos e finalmente o pseudocódigo do algoritmo de escalonamento Flow.

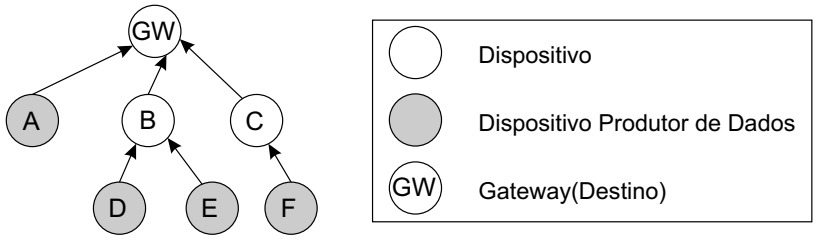
A definição de grau adotada aqui é a mesma definição clássica utilizada em estruturas de dados como árvores. No nosso caso, grau será a quantidade de saltos (transmissões) que devem ocorrer para que um dado pacote saia da origem e chegue ao seu dispositivo de destino. A Janela de Transmissão é um conceito baseado na ideia de janelas de transmissão apresentadas em [Saifullah et al. 2010]. No caso do algoritmo Flow, a Janela de Transmissão é a diferença entre o primeiro *slot* no qual o primeiro *link* do fluxo pode ser escalonado e o último *slot* no qual a última transmissão do fluxo pode ser escalonada, respeitando-se a taxa de atualização do dispositivo que produziu o pacote.

A Criticalidade é uma métrica proposta nesta tese, que mede a quantidade de ocorrências de cada dispositivo dentre o conjunto de fluxos a serem escalonados. Desse modo, nós presentes em vários fluxos possuem uma carga maior de transmissões a serem reali-

zadas e, portanto, recebem prioridade de escalonamento. O cálculo da Criticalidade (C_{F_k}) do fluxo F_k é mostrada na Equação 4.1

$$C_{F_k} = \sum O_i / i \in F \tag{4.1}$$

Onde O_i corresponde à quantidade de ocorrências do dispositivo i em todos os fluxos a serem escalonados. O dispositivo i faz parte do conjunto de todos os nós N. O conjunto de todos os fluxos a serem escalonados é representado por F e F_k representa o k-ésimo fluxo do conjunto. Um exemplo do cálculo da Criticalidade para uma topologia simples é mostrado na Figura 4.2.



Dispositivo	Ocorrências
A	1
B	2
C	1
D	1
E	1
F	1
GW	4

Fluxo	Links	Criticalidade
F1	A->GW	1+4=5
F2	D->B->GW	1+2+4=7
F3	E->B->GW	1+2+4=7
F4	F->C->GW	1+1+4=6

Figura 4.2: Exemplo de cálculo da Criticalidade em uma topologia simples.

A Criticalidade de cada fluxo é calculada pela soma das ocorrências de cada dispositivo presente no caminho percorrido pela mensagem desde a origem até o seu destino. No caso, os fluxos vão dos respectivos dispositivos origem até o *gateway*. No exemplo da Figura 4.2, se a Criticalidade for o valor de decisão da prioridade, os fluxos F2 e F3 seriam escalonados primeiro.

O esquema de decisão utilizado no algoritmo de escalonamento Flow faz a combinação entre os três critérios apresentados (Grau, Criticalidade e Janela de Transmissão). O modo de decisão é ilustrado na Figura 4.3. Este modo é aplicado quando se comparam dois fluxos para se determinar quais deles têm a maior prioridade.

Na nossa proposta, é suporta a seleção de um, dois ou três critérios de seleção (res-

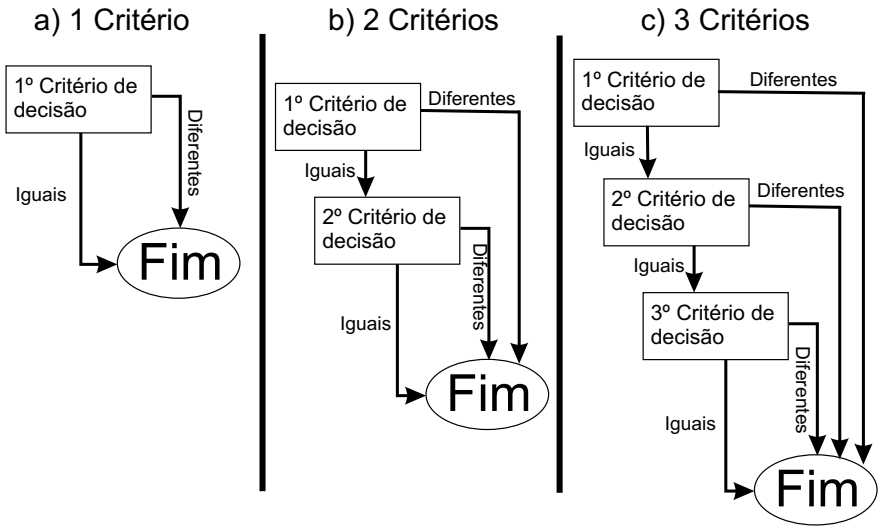


Figura 4.3: Modelos de decisão utilizados no algoritmo Flow.

pectivamente os casos a, b e c apesentados na Figura 4.3). Na situação *a*, caso o primeiro critério defina que os fluxos comparados são diferentes, esta informação é retornada, caso sejam iguais é retornado diretamente que são iguais. Nos casos com mais de um critério de decisão, caso um critério não determine que um dos fluxos é mais prioritário que os outros, esta decisão é repassada ao critério seguinte. Caso o empate persista até o último critério, o algoritmo determina que estes objetos têm a mesma prioridade.

4.2 Algoritmo Flow de escalonamento de mensagens

Nesta seção é detalhado como funciona o algoritmo de escalonamento Flow através da apresentação do pseudocódigo do mesmo.

O Algoritmo 1 corresponde à parte principal do algoritmo de escalonamento Flow. Este algoritmo recebe a lista Fluxos a serem escalonados F, o conjunto de dispositivos produtores de informações D_o e respectivas taxas de atualização destes produtores. O Algoritmo Flow retorna uma matriz M de dimensão 15 (número de canais disponíveis no *Wireless*HART) por l_M, que corresponde a quantidade de slots de tempo que equivale a duração da maior taxa de atualização (PR_{Max}) dentre os dispositivos produtores de informação. Cada elemento da matriz corresponde a uma combinação de canal com um *slot* de tempo e é preenchida por um *link* (origem, destino e dispositivo inicial de uma transmissão). Um exemplo pode ser visto na Figura 4.5.

Algoritmo 1: Algoritmo de escalonamento de mensagens Flow.

Entrada: Lista Fluxos a serem escalonados F e Conjunto de dispositivos produtores de informações D_o com valores da taxa de atualização;
Saída : Matriz M_{16xl_M} escalonada;
M// Superframe com o tamanho da maior taxa de publicação dentre os dispositivos produtores de dados;
TMSF=PR_Max/10 ms // Calculo do Tamanho do Maior Superframe ;
Chamar Gerar_Multiplos_Fluxos();
for *Todos os slots t, sendo 0<=t<TSF* **do**
  while *t<TMSF e (ainda há fluxos não escalonados em F)* **do**
    Ordenar Fluxos F de acordo com o esquema de ordenamento escolhido;
    Gerar o conjunto dos Fluxos Ativos F';
    // F': Conjunto dos fluxos em que F_i.Liberação<= t <F_i.Deadline e que NÃO estejam finalizados;
    for *todo F'i ∈F'* **do**
      //Ser escalonável significa que nenhum dos dispositivos envolvidos na transmissão (origem e destino) estejam escalonados neste slot em outro canal (Seja como origem ou destino) e que exista um canal livre para a alocação.;
      if *F'_i.ProximoHop for escalonável no slot t* **then**
        Escalonar em M F'i.ProximoHop em um canal ch disponível no slot t ;
        Atualizar Fluxo F;
        if *A alocação NÃO for bem sucedida* **then**
          Retorne "Erro de Alocação";
        end
      end
    end
    t++;
  end
end

Em seguida é chamada a função *Gerar_Multiplos_Fluxos*, que tem por objetivo adaptar os fluxos com taxas de atualização menores que a taxa de atualização máxima que define o *superframe* representado pela matriz M. Essa adaptação consiste em calcular quantas vezes o fluxo tem de ser repetidamente alocado dentro da matriz M, uma vez que vários pacotes podem ser produzidos no tempo em que os dispositivos com maior taxa de atualização produzem um único pacote. É importante notar que os tempos de liberação e de deadline das mensagens devem ser adaptados para que os *slots* reservados possam ser utilizados e não fiquem ociosos. O Algoritmo 2 representa a Função *Gerar_Multiplos_Fluxos*. Ele inicia com o conjunto de todos os Fluxos F e todas as taxas de atualização configuradas nos dispositivos produtores de dados. A saída deste algoritmo é o conjunto de fluxos F atualizado com os novos fluxos criados a partir dos fluxos com taxas de atualização menores. Para cada dispositivo produtor de dados D_o, cuja taxa de atualização seja igual a maior dentre os dispositivos, é criado um fluxo com liberação igual a zero e o deadline igual ao último *slot* do maior *superframe*. Este Fluxo criado é adicionado ao conjunto F.

Caso o dispositivo possua uma taxa de atualização menor que a taxa máxima, é calculado um fator K (número inteiro positivo) que significa quantas vezes aquele fluxo deve ser escalonado dentro de M. Assim, são criados K fluxos que têm sua liberação e seu *deadline* alterados de modo que cada fluxo tenha uma janela de transmissão com a duração igual a sua taxa de atualização. Estas janelas são definidas de modo que elas não se sobreponham. Ao final os fluxos criados são adicionados a F. Após a chamada à função da *Gerar_Multiplos_Fluxos*, o algoritmo principal é retomado e itera todos os *slots* de tempo t enquanto ainda existam fluxos para serem alocados. A cada iteração, os fluxos são ordenados de acordo com a prioridade previamente escolhida. A partir deste conjunto de fluxos ordenados, é criado o subconjunto dos fluxos ativos F'. F' consiste nos fluxos que estão dentro de sua janela de transmissão e que ainda não foram finalizados.

Para cada item F'_i, é verificado se o próximo *link* pertencente a F'_i é escalonável naquele *slot* de tempo t. Ser escalonável significa que nenhum dos dispositivos envolvidos na transmissão (origem e destino) estão escalonados neste *slot* de tempo em outro canal (seja como origem ou destino) e que exista pelo menos um canal livre para a alocação naquele *slot* de tempo. Em caso positivo, o *slot* é escalonado no *slot t* e canal *ch* e o fluxo em questão é atualizado em termos de qual o próximo *link* será escalonado ou se o fluxo foi finalizado. Em caso negativo, o *link* deste fluxo não é escalonado e o algoritmo segue para o próximo fluxo F'_i. Terminado os fluxos em F', o algoritmo incrementa o *slot t* e prossegue até que todos os *links* de todos os fluxos sejam alocados ou que t ultrapasse o número de *slots* disponiveis.

Algoritmo 2: Algoritmo da Função *Gerar_Multiplos_Fluxos*.

Entrada: Conjunto Fluxos a serem escalonados F e Conjunto de dispositivos produtores de informações D_o com valores da taxa de atualização;
Saída : Conjunto ATUALIZADO de fluxos a serem escalonados F;
// PR_i Taxa de Publicação dispositivo i;
// MPR A maior dentre as taxas de Publicação dentre os dispositivos que produzem dados (iniciais) nos Fluxos;
for *Para cada Dispositivo Produtor de dados* D_o *pertencente a* F **do**
    if $PR_{D_o} < MPR$ **then**
        K = MPR/PRD_o // K é Número de repetições do fluxo ;
        $TSFPR_i = PR_i/10$ ms //Tamanho do superframe baseado em PR_{D_o};
        for $i = 0$ *até* K **do**
            Criar Novo Fluxo NF a partir dos caminho entre D_o e o destino fornecido pelo roteamento;
            NF.Liberação = K*$TSFPR_i$;
            NF.Deadline = (K*$TSFPR_i$+($TSFPR_i$-1);
            Adicionar NF a F
        end
    else
        Criar Novo Fluxo NF a Partir dos caminho entre D_o e o destino fornecido pelo roteamento;
        NF.Liberação = 0 // O Release é o primeiro slot do maior superframe;
        NF.Deadline = ($MPR/10ms$)-1 // O deadline é o Último slot do maior superframe;
        Adicionar NF a F;
    end
end

4.3 Resultados

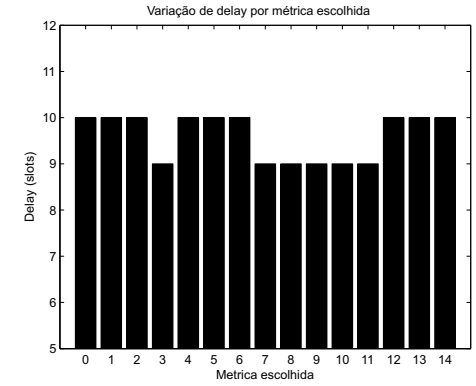
Neste seção vamos descrever os experimentos de validação do algoritmo de Escalonamento de mensagens Flow e os resultados obtidos pela execução dos mesmos. Foram realizados 5 experimentos e seus objetivos e metodologias são descritos nas seções seguintes. É importante salientar que todos os testes foram realizados utilizando-se a plataforma de simulação desenvolvida no Capítulo 5. Para propósito de comparação de desempenho, os resultados obtidos pelo algoritmo Flow são confrontados aos obtidos pelo algoritmo de Han [Han et al. 2011*b*]. As métricas de desempenho utilizadas foram as seguintes:

- O *Delay* médio – consiste no valor médio da quantidade de *slots* necessários para que sejam escalonados todos os *links* que o formam cada fluxo vindo de cada dispositivo de origem.
- Taxa de sucesso de alocação de *links* – é a razão entre a quantidade de *links* que foram escalonados de fato e a quantidade de *links* que deveriam ser escalonados para que todos dispositivos completassem a comunicação com o *gateway*, incluindo-se rotas alternativas.
- Taxa de conclusão de fluxo – considerando-se como fluxo o conjunto de *links* para que se leve um pacote de um dispositivo produtor até o *gateway*, a taxa de conclusão de fluxo é a razão entre o número de fluxos com todos os *links* escalonados e o número total de fluxos.
- Taxa de Alocação do *superframe* – é a razão entre a quantidade de *links* escalonados e o número de *slots* disponíveis dentro do *superframe*.

4.3.1 Teste de validação do algoritmo Flow

Este experimento objetiva analisar uma das principais características do algoritmo Flow que é a possibilidade de se selecionar e combinar três critérios para a escolha do *link* a ser escalonado, o que proporciona uma maior flexibilidade no uso. A Figura 4.4 mostra a tabela de critérios utilizados, indicando através de números identificadores quais são utilizados e em qual ordem de prioridades eles são utilizados. Esta configuração interfere diretamente na ordenação da prioridades de fluxos mostrados no Algoritmo 1, determinando quem tem a maior prioridade na comparação direta entre dois fluxos.

O gráfico de barras apresentado na Figura 4.4 representa o *delay* calculado em quinze experimentos de simulação (um experimento por barra) utilizando a topologia Yang [Yang et al. 2015] (mostrada na Figura 4.9) com apenas os nós folhas produzindo informações com taxa de atualização de 250 ms. Para cada experimento é utilizado um critério de



Citérios de Decisão:
G - Grau
J - Janela de Transmissão
C - Criticalidade

*Critérios apresentados em ordem decrescente de prioridade.

Identificador	Critérios*
0 (Default)	G -> J -> C
1	G
2	J
3	C
4	G->J
5	G->C
6	J->G
7	J->C
8	C -> G
9	C -> J
10	C -> J -> G
11	C -> G -> J
12	J -> G -> C
13	J -> C -> G
14	G -> C -> J

Figura 4.4: Tabela de critérios e gráfico de *delay* médio para variação de critérios em uma topologia de rede Yang.

decisão identificado no eixo horizontal e mostrado na tabela da Figura 4.4. Os critérios de decisão funcionam como o modelo apresentado na Figura 4.3. Há uma visível variação do *delay* médio entre os experimentos devido apenas aos diferentes critérios de decisão utilizados. É possível notar também que os experimentos que utilizam o critério da Criticalidade como primeiro critério (identificadores 3 e de 8 a 11) e o identificador 7 (que a utiliza como segundo critério) obtiveram um melhor desempenho quando comparados ao demais.

A Figura 4.5 mostra como em uma mesma topologia, as diferentes prioridades de fluxos selecionadas podem afetar diretamente no escalonamento gerado.

Os nós marcados em cinza são produtores de dados com taxa de atualização de 500 ms com exceção do nó 7, que está configurado com taxa de atualização 250 ms. Na Figura 4.5, o primeiro escalonamento mostrado no item **a** utiliza o grau do nó como critério de escalonamento, dando prioridade aos dados produzidos no dispositivo 2. No caso do item **b**, o critério utilizado é a janela de transmissão e como o nó 7 tem uma taxa de atualização menor, os pacotes produzidos nesse dispositivo devem ser entregues em no máximo 250 ms, enquanto os pacotes originados nos demais dispositivos produtores têm até 500 ms para serem entregues ao *gateway*. Por fim no item **c** , os dispositivos ligados diretamente

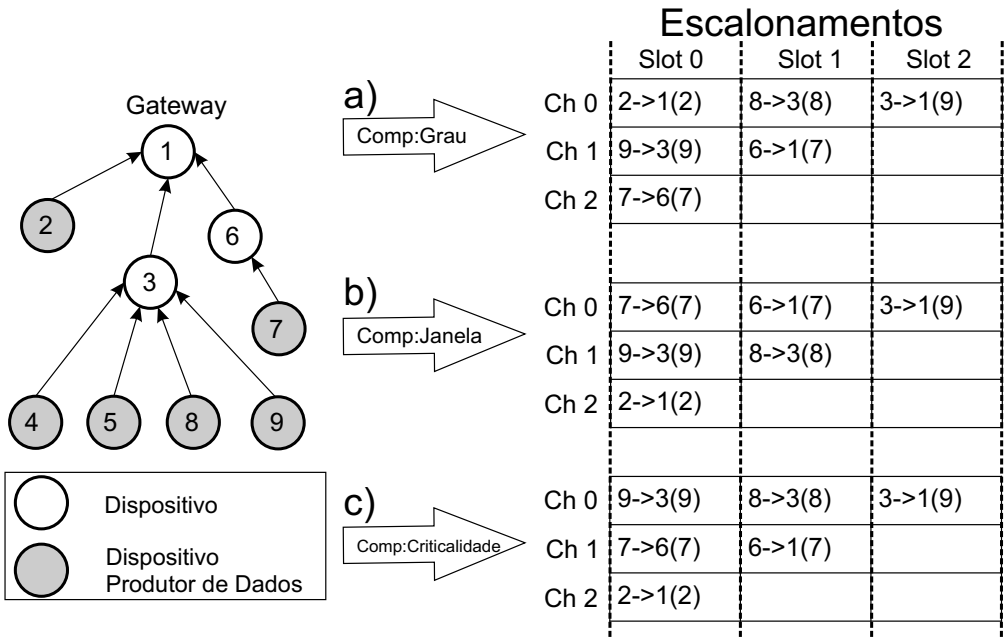


Figura 4.5: Impactos da variação de critério de comparação em uma mesma topologia.

ao nó 3 têm prioridade maior quando se refere à Criticalidade, uma vez que todos estes nós (4, 5, 8 e 9) utilizam o nó 3 como caminho ao *gateway*, aumentando a Criticalidade dos fluxos que utilizam esta rota até o *gateway*. No caso do exemplo do item **c**, o algoritmo priorizou o *link* do dispositivo 9.

4.3.2 Análise de escalabilidade para número de dispositivos

Este experimento visa verificar o impacto do aumento no número de estações produtoras de informação na rede. Serão analisados dados de *delay* médio de cada fluxo, a taxa de sucesso de alocação do *links*, a taxa de conclusão de fluxo e a taxa de ocupação do *superframe*. São comparados os desempenhos do Algoritmo Flow e o Algoritmo de escalonamento Han detalhado na seção 3.2.4. Como se trata de um experimento de escalabilidade serão utilizadas topologias numeradas de G1 a G7, conforme mostrado na Figura 4.6.

Primeiramente existe a topologia base mostrada na Figura 4.6.a. Nesta topologia todos os dispositivos produzem dados que são encaminhados ao *gateway*. O dispositivo **E** é ligado diretamente ao *gateway* e todos os dispositivos utilizam somente rotas primárias a exceção do dispositivo **B**. Para provocar o impacto necessário para se avaliar a escalabilidade, a topologia representada como G1 na Figura 4.6.b é acrescida de uma nova topologia base a cada novo grupo. De tal modo, G1 tem uma única topologia base ligada

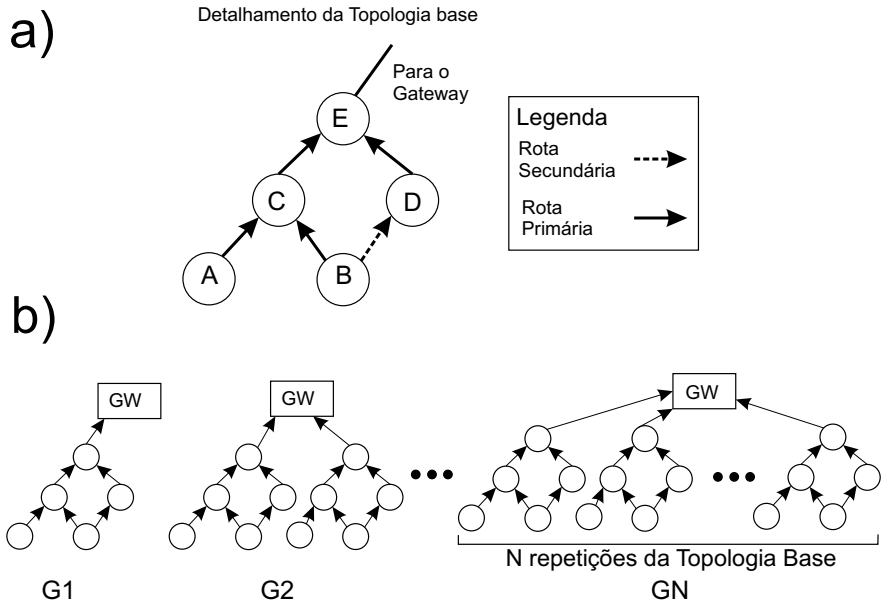


Figura 4.6: Topologia incremental para teste de escalabilidade. a)Detalhamento de topologia base. b) Progressão incremental da topologia.

ao *gateway*, G2 possui 2 topologias base ligadas ao *gateway* e assim por diante. A cada variação de um grupo ao seguinte são adicionadas 5 novas estações produtoras de informação de modo a se evidenciar o impacto nas variáveis monitoradas. Os experimentos foram realizados utilizando os Grupos G1 ao G7 com duas variações de taxa de publicação. No primeiro teste, todos os dispositivos possuem a mesma taxa de publicação fixada em 250 ms, no segundo teste os dispositivos **C** e **D** (Grau 2) de cada topologia base adicionada possuem uma taxa de atualização de 500 ms enquanto os restantes continuam com a taxa de atualização fixa em 250 ms. Os resultados são apresentados na Figura 4.7 para o primeiro teste (taxa de atualização dispositivos fixa em 250 ms), enquanto na Figura 4.8 são apresentados os resultados para o segundo experimento (taxa de atualização mista entre 250 ms e 500 ms conforme descrito anteriormente).

Pela Figura 4.7 é possível notar que em todas as variáveis monitoradas o comportamento dos resultados é semelhante entre os algoritmos Flow e Han. Podemos notar também que nas taxas de sucesso de alocação de *links* e de conclusão de fluxo se mantém constantes em 100% até o experimento G5. Neste experimento, o número de *links* que devem ser alocados ultrapassa a capacidade do *superframe* correspondente a taxa de atualização de 250 ms, causando a diminuição verificada nestas taxas de sucesso e conclusão. Outro fato importante é que o algoritmo de escalonamento Flow apresenta um desempenho melhor quando comparado ao algoritmo Han nas quatro variáveis analisadas. O algoritmo Flow apresentou menor *delay* médio, taxa de ocupação do *superframe*, taxas de

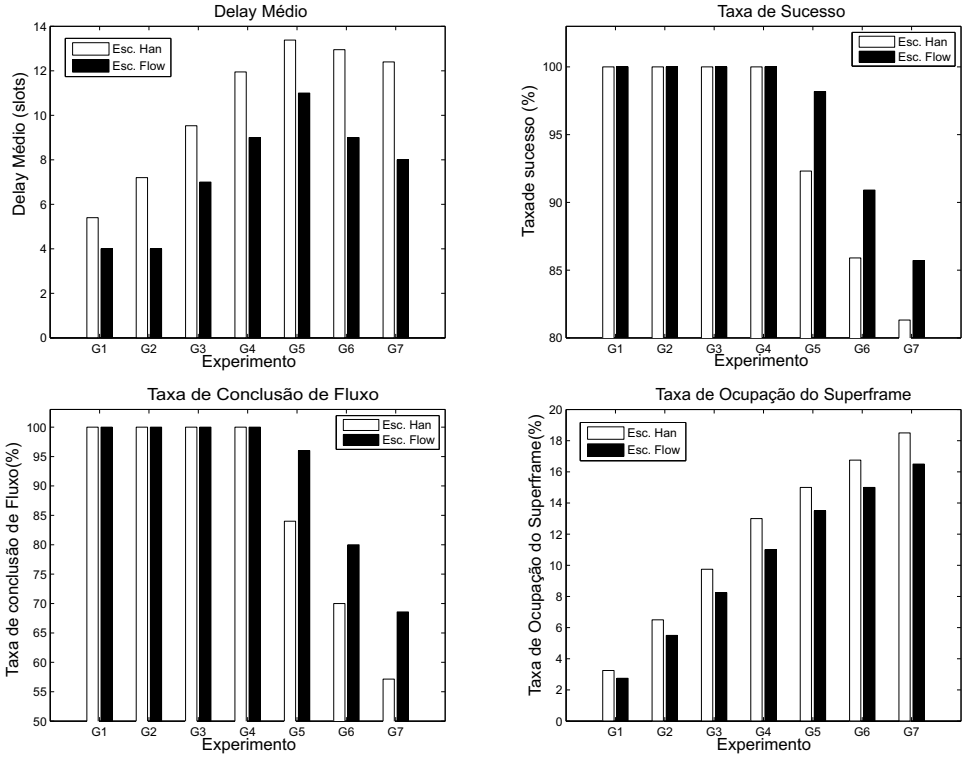


Figura 4.7: Resultados do teste de escalabilidade para taxa de atualização 250 ms.

sucesso de alocação de *links* e de conclusão de fluxos.

Já para o experimento 4.8, a semelhança do comportamento assintótico entre os algoritmos de escalonamento é mantida. Embora o escalonamento Flow tenha conseguido um desempenho superior ao algoritmo Han, a diferença entre o desempenho de ambos foi reduzida quando comparada ao experimento com uma única taxa de atualização. Isso pode ser justificado pelo fato de que o algoritmo Han tem um melhor desempenho quando lida com várias taxas de atualização, uma vez que a mesma é utilizada como métrica pelo algoritmo como visto na Seção 3.2.4. Podemos verificar também que a queda de desempenho das taxas de sucesso se inicia no experimento G5 para ambos os algoritmos, embora essa queda seja menor do que a observada no mesmo experimento da Figura 4.7 com taxa de atualização 250 ms. Podemos atribuir isso ao fato de que, por utilizar uma taxa de atualização maior, o *superframe* para alocação também deve ser maior para representar essa periodicidade da coleta de dados.

4.3.3 Análise de impacto da variação do número de dispositivos

Este experimento visa verificar o impacto do aumento no número de estações produtoras de informação na rede para a topologia apresentada na Figura 4.9. Esta topologia foi retirada do trabalho apresentado em [Yang et al. 2015] e representa um cenário industrial

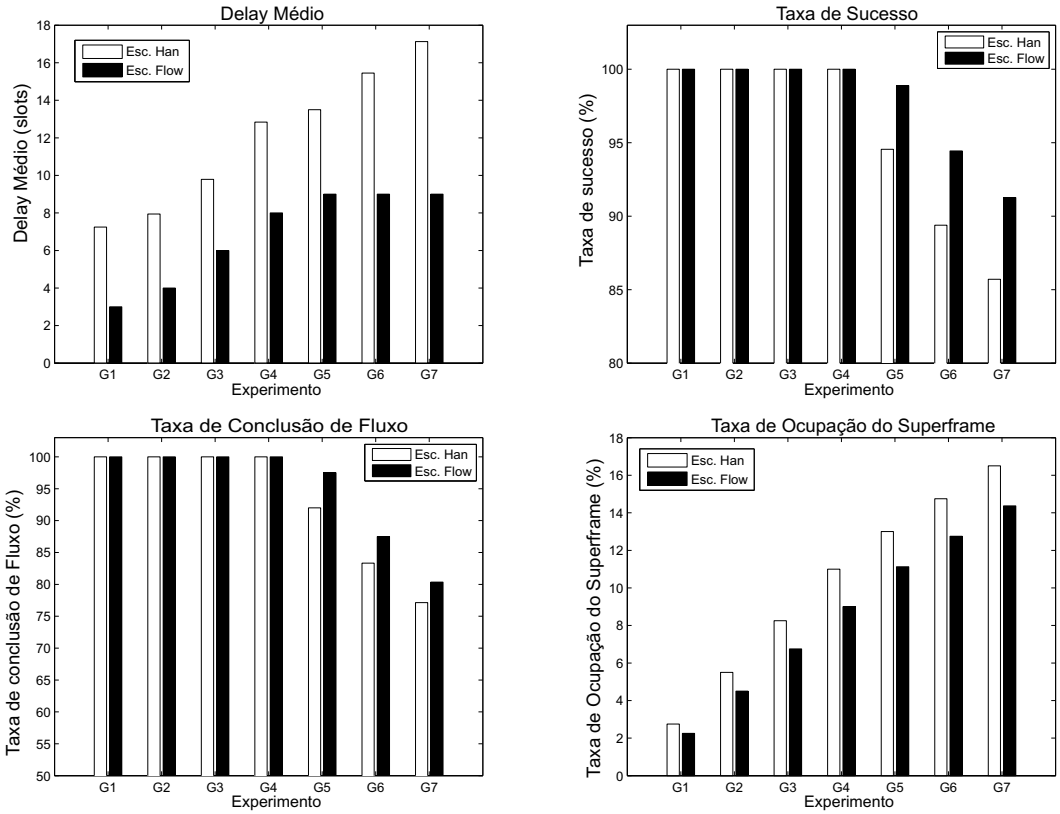


Figura 4.8: Resultados do teste de escalabilidade para taxas de atualização 250 ms e 500 ms.

de produção típico de sistemas de soldagem.

Serão analisados novamente dados de *delay* médio de cada fluxo, taxa de sucesso de alocação de *links*, taxa de conclusão de fluxo e taxa de ocupação do *superframe*. São comparados os desempenhos do algoritmo Flow e o algoritmo de escalonamento Han. Neste experimento todos os dispositivos "folha"estão configurados para produzir informações com taxa de atualização de 250 ms e destinadas ao *gateway*. São utilizadas apenas rotas primárias na comunicação entre os dispositivos de produtores de dados e o *gateway*. O teste é realizado aumentando-se o número de estações produtoras de informação. A quantidade de dispositivos varia entre 24 a 29 dispositivos, como mostrado nos resultados da Figura 4.10.

Pelos resultados obtidos, notamos que neste cenário com apenas rotas primárias o algoritmo Han obteve um melhor desempenho do que o Flow em termos de taxa de sucesso de alocação de *links* e de conclusão de fluxos. Contudo, o algoritmo Flow continua com melhor desempenho em termos de *delay* médio e Taxa de ocupação de *superframe*. Destaca-se também que o algoritmo de Han consegue manter 100% de alocação de *links* e de conclusão de fluxos ainda com 26 dispositivos produtores de dados, enquanto o

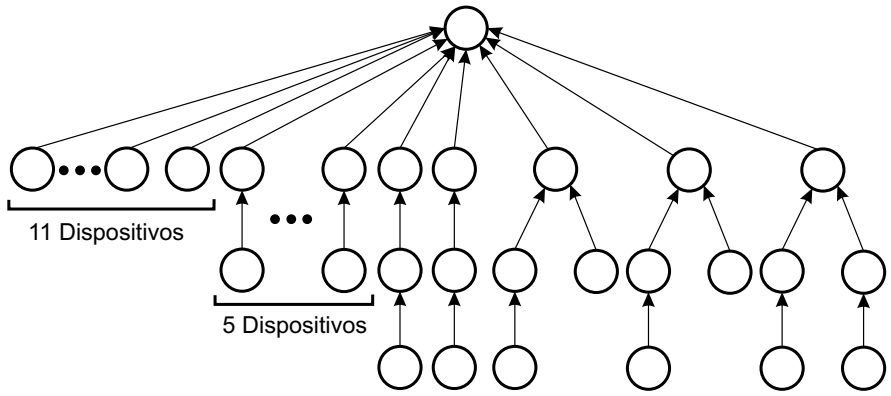


Figura 4.9: Topologia um cenário industrial de produção típico de sistemas de soldagem.

algoritmo Flow mantém esta condição somente até o experimento com 25 dispositivos produtores de dados para esta topologia.

4.3.4 Análise de impacto do aumento da taxa de atualização de dispositivos

Este experimento foi inspirado no experimento realizado em [Han et al. 2011*b*], e visa analisar o impacto dos diferentes valores de taxa de atualização suportados pelo *Wireless*HART em uma mesma topologia. Para este experimento é utilizada a mesma topologia apresentada em Figura 4.6, na configuração G10, com 50 dispositivos produtores de informação configurados com a mesma taxa de atualização e que enviam dados ao *gateway*.

Os parâmetros monitorados são *delay* médio de cada fluxo, taxa de sucesso de alocação dos *links*, taxa de conclusão de fluxo e a taxa de ocupação do *superframe*. Os resultados podem ser vistos na Figura 4.11.

É importante ressaltarmos que cada aumento de taxa de atualização influencia diretamente no tamanho disponível para alocação de *links* no *superframe* uma vez que este último é calculado para que possa acomodar todas as taxas de atualização disponíveis na rede. Como estamos tratando com o protocolo *Wireless*HART, as taxas de atualização estão restritas a um conjunto de valores predefinidos como mostrado no Capítulo 2. Inicialmente, o *delay* médio tem valores menores, uma vez que se são contabilizados apenas os fluxos escalonados e existe uma quantidade de *links* não escalonados devido ao tamanho do *superframe* para os primeiros valores de taxa de atualização. Porém, isso vai sendo modificado até que é alcançado o valor de 1s de taxa de atualização. Ainda em termos de *delay*, é notada uma leve vantagem para o algoritmo de escalonamento Flow sobre o Han.

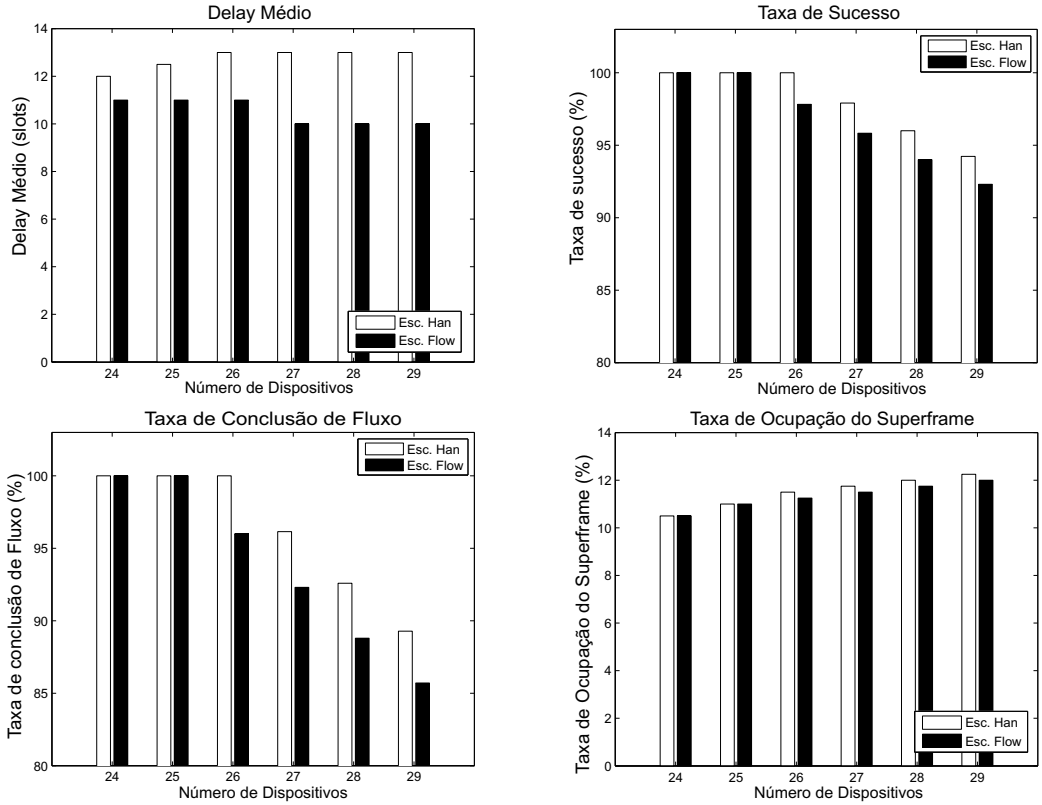


Figura 4.10: Resultados do aumento de dispositivos em um cenário industrial de produção típico de sistemas de soldagem

A mesma relação entre os primeiros valores e o tamanho do *superframe* também explica o comportamento da taxa de sucesso de alocação de *links* e de conclusão de fluxos.

Nos primeiros casos de teste a falta de *slots* para a alocação de *links* implica na diminuição da taxa de alocação de *links* para a faixa dos 70% em ambos os algoritmos e a taxa de conclusão de fluxo para valores entre 40% e 60%. O desempenho superior do algoritmo Flow em termos de taxa de sucesso de alocação de *links* se mantém em relação ao algoritmo Han nas duas primeiras taxas de atualização, até quando ambas atingem o 100% no teste de 1s. Já em termos da taxa conclusão de fluxo, a vantagem do algoritmo Flow sobre o Han chega a aproximadamente 10%. Esta vantagem em termos de taxa de conclusão de fluxo se mantém em todos os testes até que ambos os algoritmos atingem 100% de taxa de conclusão de fluxo no teste com taxa de atualização de 1s. Também, após o limiar da taxa de atualização de 1s, ambas as taxas, sucesso de alocação de *links* e a de conclusão de fluxo, medidas de ambos os algoritmos, alcançam e permanecem em 100% devido à abundância de *slots* disponíveis para a alocação. Também a taxa de ocupação do *superframe* apenas diminui com o aumento da disponibilidade de *slots*.

Na maioria dos experimentos, o algoritmo Flow obteve um melhor desempenho quando

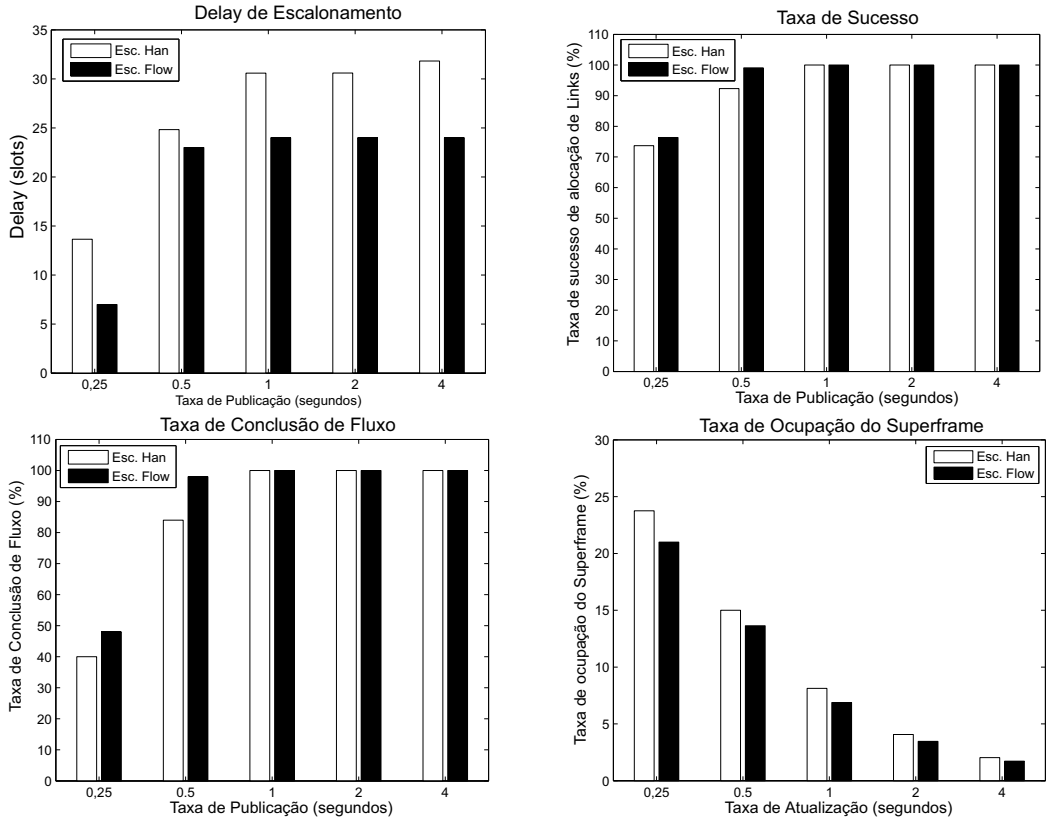


Figura 4.11: Resultados da variação na taxa de atualização de 50 dispositivos em uma topologia G10

comparado ao algoritmo Han nos critérios avaliados. Podemos destacar que o algoritmo Flow manteve um desempenho melhor ou equivalente ao algoritmo Han em todos os experimentos em termos de *delay* e de taxa de ocupação de *superframe*. A exceção ocorreu no experimento da análise de impacto da variação do número de dispositivos, em que o algoritmo de escalonamento Han obteve melhor desempenho em termos de taxa de sucesso de alocação de links e taxa de conclusão de fluxo.

Capítulo 5

Módulo de simulação para o NS-3

Uma das dificuldades encontradas para a criação e a otimização de protocolos de redes é a análise de tais protocolos em diferentes ambientes. A montagem de uma infraestrutura para testes dos diferentes ambientes com máquinas reais se mostraria custoso (embora com resultados mais precisos) com o aumento do número de estações tanto do ponto de vista financeiro quanto do ponto de vista do tempo de desenvolvimento.

Com base nesses fatos, uma solução com custos reduzidos e que atenda a flexibilidade necessária para os testes é a simulação computacional. Com a simulação pode-se obter resultados a partir de diferentes topologias e número de estações com o gasto de tempo relativo apenas a configuração dos parâmetros de simulação. Neste capítulo será mostrada a organização de um módulo de simulação para o Network Simulator 3 (com um breve detalhamento sobre este) e as suas principais características.

5.1 Network Simulator 3

O padrão atual para as simulações de redes é o simulador Network Simulator 2 (NS-2) segundo [Weingärtner et al. 2009]. O NS é um simulador de eventos discretos para pesquisas em redes, provendo recursos para simulação do protocolo TCP, roteamento e protocolos *multicast* em redes sem fios ou cabeadas, locais ou via satélites [Coutinho 2003]. O NS é programado em duas linguagens: C++ e a linguagem interpretada OTCL (*Object-oriented Tool Command Language*). A linguagem OTCL é utilizada para a configuração dos parâmetros da simulação, enquanto o C++, devido a sua robustez, maior disponibilidade de pacotes, velocidade de execução dos códigos compilados e facilidade de manipular diretamente variáveis em bytes, é a escolha para que sejam programados os protocolos a serem utilizados [Coutinho 2003]. O OTCL utiliza objetos compilados em C++ a partir de um processo de ligação (*linkage*) via TclCl (*Tcl with Classes*), relacionando-os [Demarch 2007].

O NS começou em 1989. Em 1995 o desenvolvimento do NS foi mantido pela DARPA (*Defence Advanced Research Projects Agency*) através do projeto VINT que inclui empresas como a LBL, Xerox PARC, UCB e USC/ISI. Atualmente, o desenvolvimento do NS é mantido de maneira colaborativa por diversas instituções e pesquisadores como, por exemplo, pela americana DARPA (*The Defense Advanced Research Projects Agency*). O desenvolvimento também é auxiliado por outros grupos como, por exemplo, a Sun Microsystems.

Entretanto, o NS-2 se mostra com grandes problemas com relação à escalabilidade na utilização de memória e tempo de simulação [Weingärtner et al. 2009]. O simulador também apresenta grandes abstrações nas camadas de rede e camadas mais baixas, o que traz dificuldades quanto à integração da simulação com dispositivos reais, além de que a documentação existente do NS-2 é ultrapassada ou inexistente [NS-3 Consortium 2015]. Para sanar tais dificuldades, os desenvolvedores do NS-2 começaram o projeto do Network Simulator 3 (NS-3) [NS-3 Consortium 2015].

Apesar de existirem outros simuladores, o estudo conduzido em [Weingärtner et al. 2009] apontou o NS-3 como tendo a melhor desempenho no geral. Entretanto o NS-3 ainda apresenta poucos módulos implementados, o que nos motiva ao desenvolvimento de um módulo *Wireless*HART para o simulador. Outro fator que contribui em favor do NS-3 é o de que este é um projeto de software livre, não requerendo licenças para a sua utilização.

O Network Simulator 3 é um simulador de eventos discretos desenvolvido em um projeto de código aberto sobre uma licença GNU GPLv2 iniciado em 2006 e organizado em torno da comunidade de pesquisa que o mantém, desenvolve e realiza a manutenção do mesmo. O NS-3 tem por objetivo permitir que pesquisadores estudem protocolos em sistemas em larga escala e em ambientes controlados, além de poder ser usado para fins de educacionais [NS-3 Consortium 2015]. É importante salientar que o NS-3 não é compatível com o NS-2, sendo um projeto diferenciado de seu antecessor.

O NS-3 é implementado na linguagem de programação C++, bem como os seus *scripts* de simulação, que também são classes C++. A linguagem Phyton pode ser opcionalmente utilizado pelo usuários. O simulador utiliza uma arquitetura modularizada e orientada a objeto, o que facilita o entendimento do código e a reutilização do mesmo.

Nas próximas seções serão descritas algumas estruturas básicas do NS-3. Essa revisão se faz necessária devido ao fato de que alguns jargões de redes de computadores têm um significado específico no NS-3. Todas as informações a seguir foram retiradas de [Project 2010].

5.1.1 Node

O *node* representa o dispositivo que se conecta à rede. Tal termo mais genérico é utilizado a partir da Teoria de Grafos de modo a expressar que o nó é algo mais genérico que não se relaciona diretamente com a Internet e seus protocolos (ao contrário do termo *host*, por exemplo). Logo o *node* será o abstração dispositivo computacional no NS-3, de modo que a classe Node, implementada em C++ fornece os métodos necessários para a representação dos dispositivos, como, por exemplo, a instalação de uma placa de rede. Assim o Node é como um computador ao qual o usuário adiciona funcionalidades como diferentes aplicações, protocolos e periféricos.

5.1.2 Aplicações

No NS-3 não há um conceito real de sistema operacional, chamadas de sistema ou privilégio de usuários. Logo as aplicações, que no mundo real desempenham uma tarefa no sistema, servem para gerar e direcionar as atividade na rede simulada. Esse papel é desempenhado pela classe *Application*, que provê métodos para manter as representações de aplicações em nível de usuário. Os desenvolvedores devem herdar da classe *Application* para o provimento de novas aplicações.

5.1.3 Canais

Geralmente se chama de canal no mundo real o meio pelo qual os dados são transmitidos. Por exemplo, podemos citar o ar como canal em uma transmissão sem fios ou o cabo pelo qual um computador se conecta em uma rede Ethernet convencional. No mundo simulado do NS-3, o usuário deve conectar os Nodes a objetos que representam os canais. A abstração do canal real é realizada pela classe *Channel*, logo implementando as características de cada meio, recebendo novas conexões de Nodes e gerenciando as comunicações entre os que estão integrados ao canal. Novos canais devem herdar de *Channel*, e podem modelar desde ambientes simples como fios até ambientes complexos em 3 dimensões, com obstáculos e modelos de propagação distintos.

5.1.4 Dispositivo de Rede

Para que uma máquina se conecte a uma rede no mundo real se faz necessário a existência de um periférico (embutido ou anexado ao hardware) que realize as funções de rede, chamados Placas de Interface de Rede (em inglês, *Network Interface Cards* – NICs). A

NIC juntamente com o *driver* (programa que controla a peça de hardware) fornecem as funcionalidades de rede ao sistema no qual foram instaladas. No NS-3, as abstrações que representam a NIC e o *driver* são implementadas na classe *Net Device*. Os objetos da classe *Net Device* são "instalados" nos Nodes, de modo que podem se comunicar por um canal. Um único Node pode possuir diferentes *Net Devices* e comunicar-se por diferentes canais, o que se equivale a um *notebook* que se conecta via cabo e por redes sem-fio. Logo, para se desenvolvar um novo dispositvo, o desenvolvedor deve herdar da classe *Net Device*, que provê métodos para a comunicação das classes superiores com as classes *Node* e *Channel*.

5.1.5 Topology Helpers

A criação de uma rede de dispositivos consiste em uma série de comandos em C++ para instanciar, inicializar e ligar as diversas classes Node, Channel e Net Device, por exemplo. As classes chamadas de Helpers tem por função automatizar tais comandos de modo que esta atividade se torne o mais simples possível para o usuário, podendo ser repetida, reconfigurada e ter sua escala aumentada de maneira fácil. Outros exemplos de utilizações de classes Helpers são para funções de distribuição de estações no espaço e para e de endereços de rede.

5.2 Módulo *Wireless*HART para o NS-3

A arquitetura implementada tem como principal foco o desenvolvimento de uma base sólida da camada física do protocolo e das características do canal de transmissão. As camadas superiores foram abstraídas de modo que apenas as funcionalidade de roteamento e escalonamento fossem implementadas para que se pudessem desenvolver as primeiras simulações da rede *Wireless*HART. De tal modo, o escalonamento e roteamento deve ser fornecidos como parâmetros de configuração da simulação.

Tratando-se de roteamento, o usuário deve fornecer obrigatoriamente para cada estação o número da estação que deve servir como rota principal para o encaminhamento dos pacotes. Opcionalmente, o usuário também pode inserir um número de estação para atuar como destino secundário de roteamento.

Quanto ao escalonamento, a configuração deve ser feita como se o próprio usuário do simulador realizasse a funcionalidade do Network Manager. Este escalonamento deve atender às restrições apresentadas na subseção 2.2.2 que, grosseiramente, consistem em enviar o pacote duas vezes pela rota principal e uma vez pela rota secundária (caso exista).

A estrutura para inserção de um *link* no *superframe* ocorre chamando a função *AddLink-Superframe* e fornecendo os parâmetros:

- Id do *link*;
- Estação fonte do pacote;
- Offset de Canal;
- Estação de origem da transmissão;
- Estação de destino da transmissão e
- Slot de tempo dentro do *superframe*.

A arquitetura implementada é descrita pelo diagrama de classes apresentado na Figura 5.1.

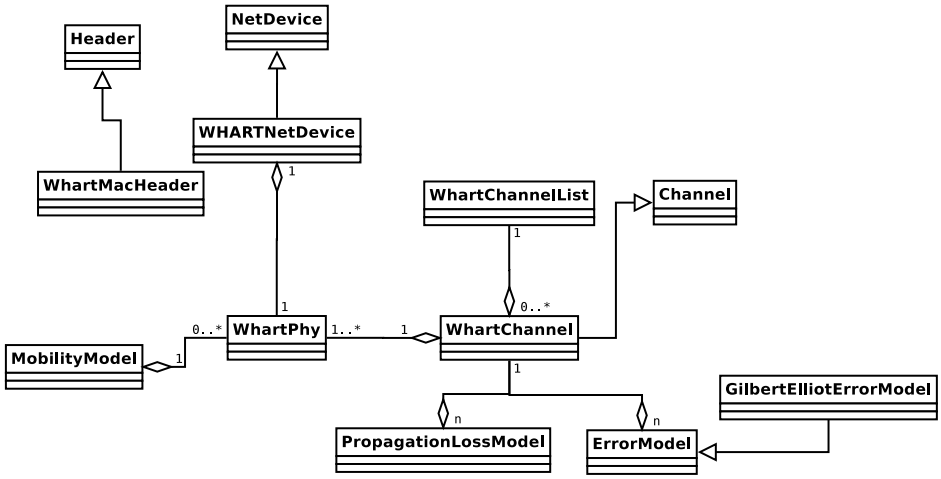


Figura 5.1: Diagrama de classe da camada física do *Wireless*HART.

As classses NetDevice e Channel são classes padrão do NS-3, como explicado na subseção 5.1, e devem ser herdadas, respectivamente, para a implementação da camada de enlace e das características do meio de transmissão. A classe WhartNetDevice representa a implementação preliminar das funcionalidades da camada de enlace do *Wireless*HART. Nessa versão simplificada, a WhartNetdevice implementa os *buffer* de transmissão, o escalonamento da rede e as informações de roteamento. Como esta camada é a de mais alto nível implementada, ela está simulando alguns comportamentos das camadas superiores, como produzir e consumir informações e guardar informações de roteamento.

O funcionamento do algoritmo desta classe está implementado de acordo com a máquina de estados TDMA apresentada na especificação *Wireless*HART [(IEC) 2010].

A classe Header é a classe padrão do NS-3 para a implementação de cabeçalhos. A vantagem de sua utilização é que já é fornecida um conjunto de ferramentas próprias para

a manipulação de cabeçalhos, o que inclui a inclusão e a remoção nos objetos da classe Packet do NS-3 bem como a leitura dos dados armazenados. A herança foi realizada para implementar os atributos do cabeçalho da camada de enlace mostrados na Figura 5.2. Os atributos inseridos na classe WhartMacHeader são serializados como campos do cabeçalho incrementando o tamanho do pacote quando anexados. Como no *Wireless*HART o tamanho do cabeçalho de enlace é de acordo com a configuração do especificador de endereços, os outros campos tem seus tamanhos devidamente configurados automaticamente de acordo com o valor do campo especificador. Por exemplo quando se determina se serão utilizados endereços no padrão EUI-64 ou no padrão modo *nickname*.

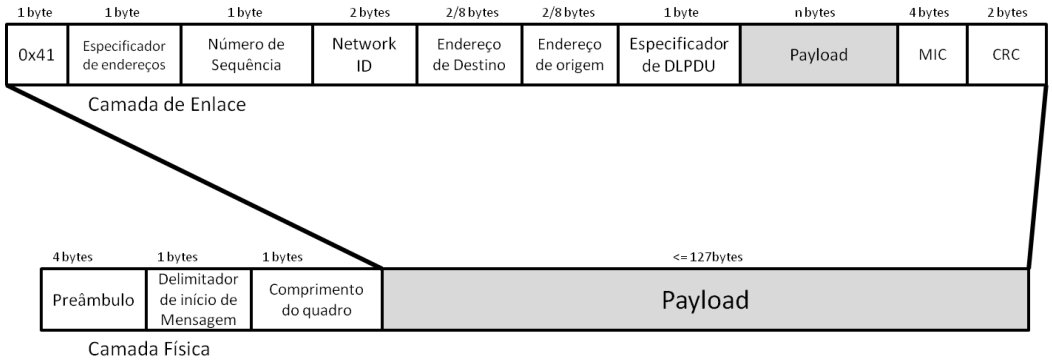


Figura 5.2: Quadro detalhado da camada de enlace do *Wireless*HART.

A implementação das classes WhartPhy, WhartChannel e suas classes correlatas são explicadas nas subseções seguintes.

5.2.1 Camada Física

A camada física é implementada na classe WhartPhy realizando a modelagem do transmissor de rádio do dispositivo simulado. Nela são incluídas características como o nível de detecção de sinais, a potência de transmissão e o modelo de movimentação do dispositivo.

Modelo do Rádio

Para estudos sobre o comportamento do consumo de energia, desenvolvemos um modelo de rádio baseado no modelo apresentado por [Ramachandran et al. 2007] a utilizando dados de consumos fornecidos em [Casilari et al. 2010]. A máquina de estados para modelar o consumo de energia é descrita na Figura 5.3.

Os estados modelados e o respectivo consumo de energia são:

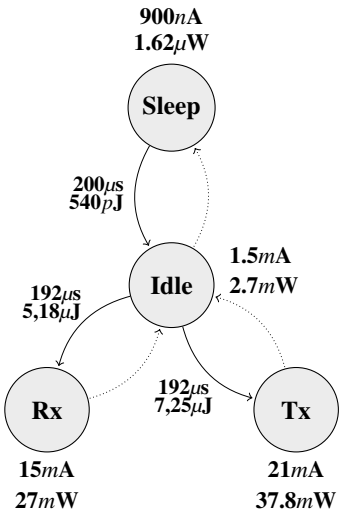


Figura 5.3: Modelo de energia

- TX: O rádio está transmitindo;
- RX: O rádio está recebendo;
- Idle: O clock está ligado e o rádio está pronto para ir para os estados Tx e Rx e
- Sleep: O clock está desligado (*clock sleep* ligado) e o rádio está aguardando a inicialização.

Todos os estados têm um acréscimo de 3,3 mA no consumo devido ao consumo do processador do dispositivo [Chen et al. 2010], exceto o estado Sleep.

Modelo de Mobilidade

O modelo de mobilidade, implementado na classe *MobilityModel* do NS-3, provê o posicionamento em três dimensões de um dispositivo, os limites espaciais (tamanho do ambiente simulado) da simulação e também pode modelar a movimentação de dispositivos nos modos velocidade constante, aceleração constante e posição fixa, por exemplo. Entretanto, a funcionalidade da mobilidade só deverá ser utilizada aqui para os dispositivos portáteis, uma vez que os outros tipos de dispositivos devem ter posicionamentos constantes de acordo com [(IEC) 2010].

5.2.2 Canal

Finalmente, a classe WhartChannel modela as características do meio e contém o modelo de propagação e modelo de erro. Para simular os 16 canais do *Wireless*HART, a classe WhartChannelList foi criada para armazenar as instâncias dos 16 canais, padronizar

o acesso a esses objetos e facilitar sua inicialização e configuração no *script* de simulação. Esta implementação favorece a flexibilidade do simulador, uma vez que cada canal pode ter modelos de erro e de propagação independentes dos outros canais.

As classes *ErrorModel* e *PropagationLossModel* são nativas do NS-3. Os modelos utilizados são explicados nas seções seguintes.

5.2.3 Modelo de Erro

A classe *ErrorModel* já provê um modelo de erro simples, entretanto para melhor representar o erro em transmissões sem fios o modelo de erro clássico de Gilbert/Elliot [Han & Lee 2007] foi implementado estendendo a classe nativa do simulador. Esse modelo calcula a taxa de perda de pacote (*Packet Error Rate* - PER) baseada na taxa de erro por bit (*Bit Error Rate* - BER). O modelo é baseado na cadeia de Markov apresentada na Figura 5.4.

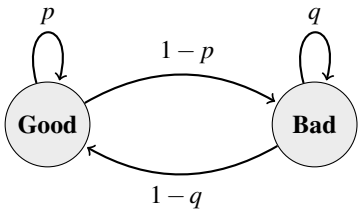


Figura 5.4: Modelo de erro de Gilbert/Elliot.

Neste modelo o estado inicial é o estado *good*. Para cada novo bit o canal pode permanecer no estado atual ou sofrer uma transição. As probabilidades de transição são mostradas na Figura 5.4, onde p é a probabilidade de que estando em um estado *good*, se mantenha nesse estado e q é a probabilidade de que estando em um estado *bad*, se mantenha nesse estado após uma transição. Por fim, as probabilidades em regime permanente (P_g e P_b, respectivamente) são calculados por.

$$P_g = \frac{1-q}{2-(p+q)} \qquad P_b = \frac{1-p}{2-(p+q)} \tag{5.1}$$

Finalmente, a equação apresentada [Han & Lee 2007] para calcular o PER para uma menssagem de n-bytes é:

$$PER(n) = 1-(P_g p^{8n} + P_b(1-q)p^{8n-1}) \tag{5.2}$$

Para uma melhor modelagem de situações reais, a classe implementada dá suporte para que cada *link* entre duas estações possua sua própria probabilidade de erro indepen-

dente da probabilidade ajustada para todo o modelo. Isso permite, por exemplo, que sejam simulados obstáculos entre determinadas estações de modo que as outros *links* não sejam afetados.

Segundo [Willig & Ebert 1999] podemos calcular o tempo que a máquina de estado de Gilbert/Elliot passa nos estados *good* (Tg) e *bad* (Tb) em regime permanente com as equações:

$$T_g = \frac{1}{1-p} \qquad T_b = \frac{1}{1-q} \tag{5.3}$$

5.2.4 Modelo de Propagação

O modelo de propagação determina as perdas de potência ocorridas durante a transmissão de sinal pelo meio, de modo que podemos calcular a potência recebida baseado no posicionamento do par receptor/transmissor e na potência do rádio transmissor. De acordo com a documentação presente em [NS-3 Consortium 2015], o NS-3 implementa nativamente, por exemplo, os seguintes modelos de propagação: Propagação constante, *Fixed Rss Loss*, Friss, *Log Distance*, Matrix, Nakagami, *Random*, *Three Log Distance* and *Two Ray Ground* [NS-3 Consortium 2015].

5.2.5 Interface com o usuário

No módulo desenvolvido, a interface amigável com o usuário para o simulador de redes tem o objetivo de facilitar a utilização do simulador sem que para isso se faça necessário recorrer a configuração manual de um arquivo do NS-3 e melhorar o entendimento do funcionamento da rede. Além disso, para o caso do módulo desenvolvido, a interface também dá suporte a implementação de algoritmos de escalonamento e de roteamento. O fluxograma deste funcionamento é mostrado na Figura 5.5 e será detalhado a seguir.

Na interface inicial mostrada na Figura 5.5a mostra o espaço para a criação da topologia. Esta é criada de maneira *drag-and-drop*, onde cada ícone representa um dispositivo (gateway, Access Point, ou dispositivo de campo) e os *links* são estabelecidos entre os dispositivos representados por setas. É possível atribuir nomes aos dispositivos para uma melhor identificação. As topologias criadas podem ser salvas em arquivo para posterior carregamento e execução, facilitando a repetibilidade de topologias e a criação de simulações de maior escopo.

Após dispostos os *links* e os dispositivos, o usuário pode configurar manualmente cada rota primária e secundária da rede através da interface *drag and drop* mostrada na

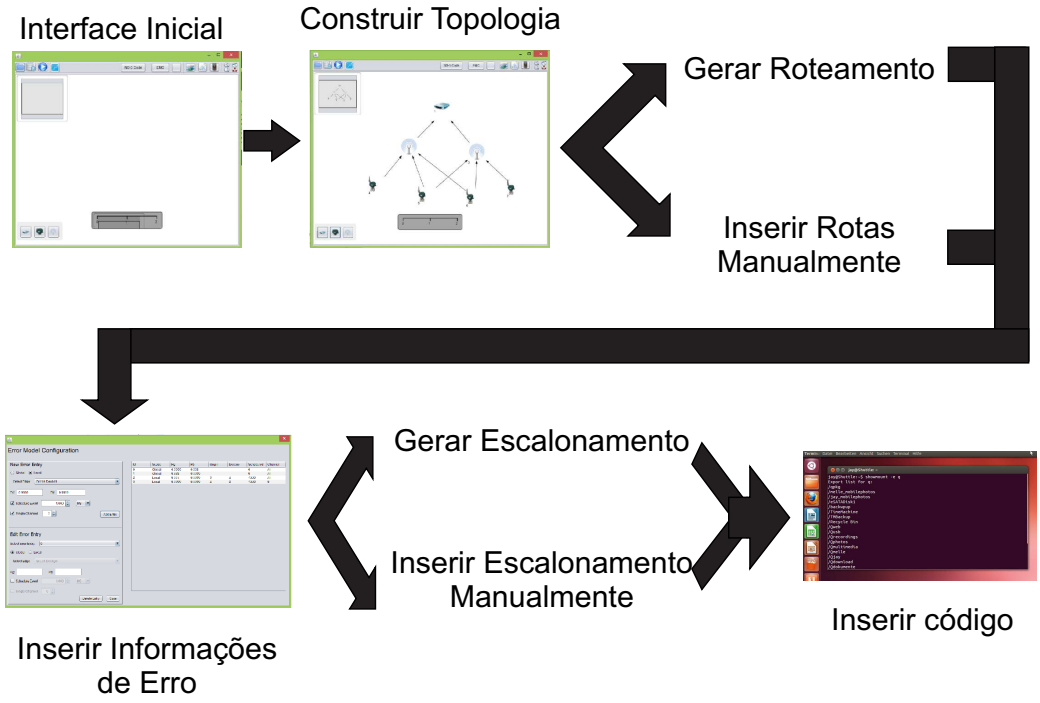


Figura 5.5: Fluxograma da interface com o usuário do módulo de simulação

Figura 5.6 ou executar um algoritmo de roteamento previamente programando. Embora o código possa receber novos algoritmos de roteamento para a geração automatizada de rotas, a implementação está utilizando apenas um algoritmo de busca em profundidade em grafos para gerar exemplos de rota.

Uma importante funcionalidade do módulo é a configuração dos parâmetros de erro da simulação. A interface permite que o usuário insira probabilidades de erro (Pg e Pb conforme o modelo Gilbert/Elliot) individuais ou globais, iniciais ou agendadas e em um único canal ou por todos os canais de um *link*. Esta funcionalidade é vital para o módulo uma vez que esta permite a variação de cenários de erro durante uma mesma simulação. As probabilidades de erro inseridas podem ser editadas caso se faça necessário. A interface pode ser vista na Figura 5.7, ilustrando as configurações possíveis de serem utilizadas.

Para o escalonamento, a ferramenta possui a geração automática através de algoritmos previamente programados (algoritmos Flow, Han e First Fit) ou uma ferramenta de criação manual do *superframe*. Na geração automática, a partir das rotas estabelecidas e da definição dos nós produtores de informação (sensores), o sistema gera o *superframe* a ser utilizado durante a simulação. O grafo de roteamento pode ser inserido através de uma interface *drag and drop*, especificando as rotas alternativas quando necessário. Quanto

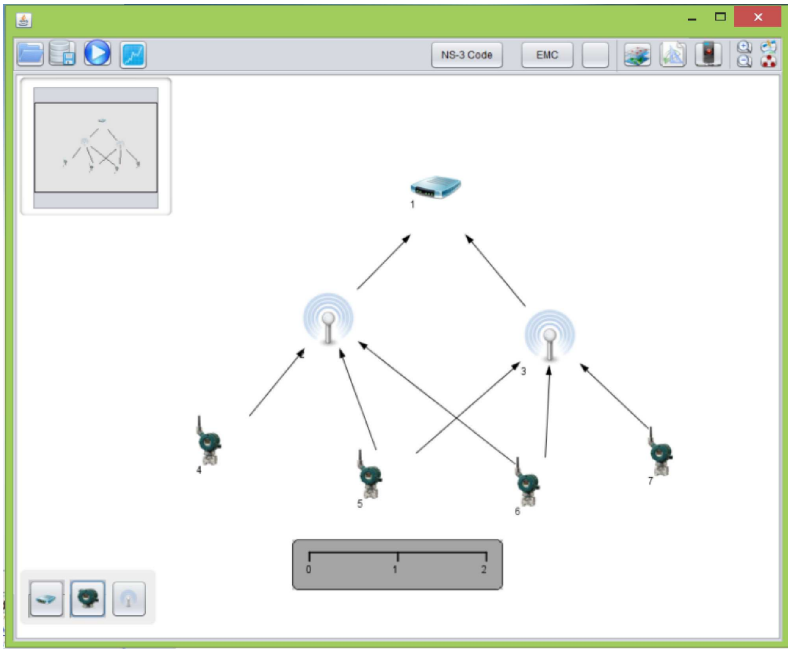


Figura 5.6: Interface de entrada de grafo de roteamento.

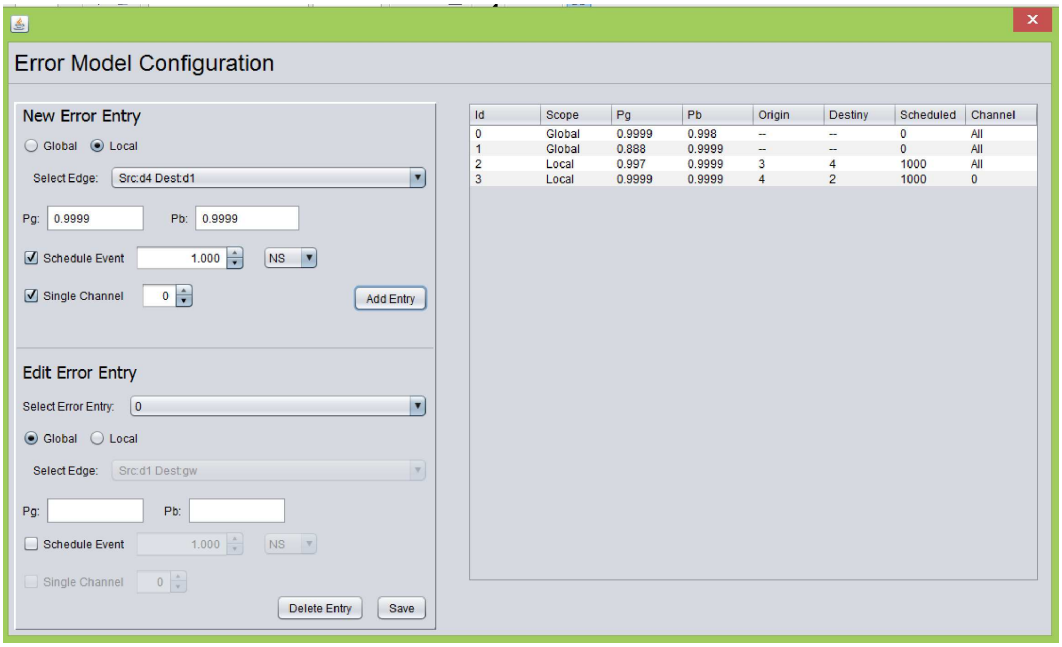


Figura 5.7: Interface de configuração de probabilidade de erro.

a ferramenta de criação manual do *superframe* (independente de algoritmo), esta tem a função de se inserir cada transmissão no *superframe*, possibilitando inclusive a implementação (embora mais trabalhosa) diferentes algoritmos não implementados no sistema para

efeito de comparação. Uma ilustração de como a configuração e escalonamento é feita pode ser vista na Figura 5.8 e com destaque para o modo de visualização dos *superframes* criados.

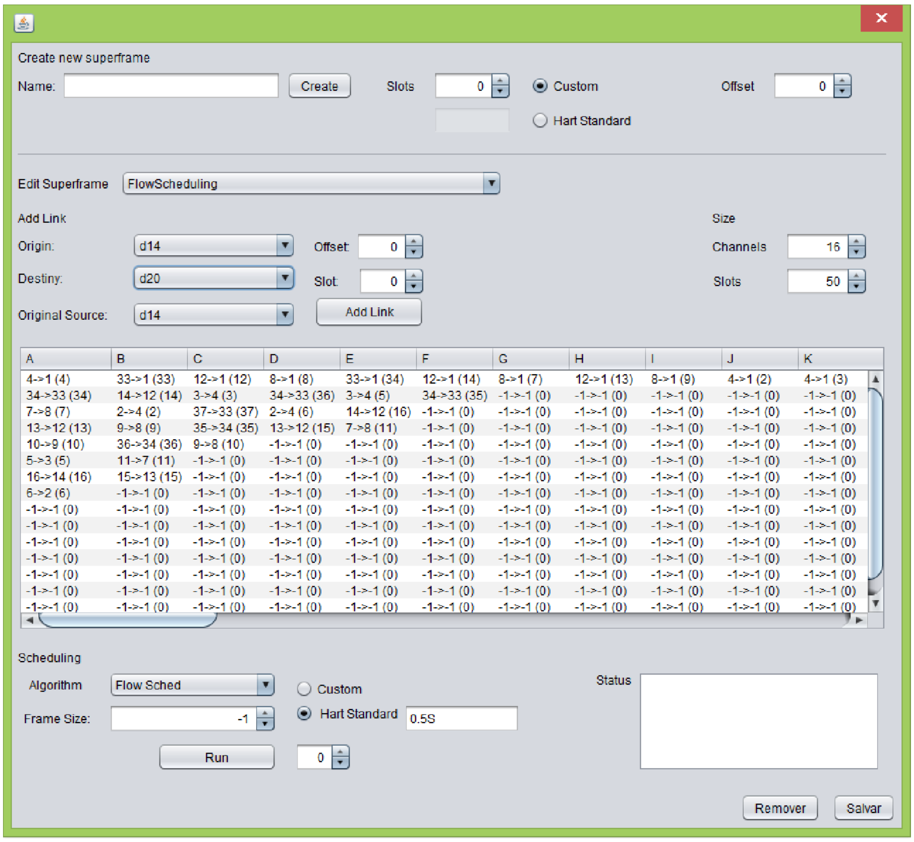


Figura 5.8: Interface de configuração de escalonamento.

Finalmente, concluídos os passos anteriores, o programa pode realizar a geração de código para o NS-3 a partir das configurações realizadas. Ao realizar este comando o usuário define o arquivo de saída e este arquivo pode ser executado diretamente no NS-3. O funcionamento da interface é independente do NS-3 e os *superframes* gerados podem ser utilizados como entrada para outros simuladores ou para comparações entre algoritmos. É importante citar também que esta interface foi programada em linguagem java e pode ser utilizada em qualquer sistema operacional que suporte esta tecnologia.

Capítulo 6

Resultados da análise de desempenho

Neste Capítulo serão apresentados os resultados obtidos utilizando o módulo de simulação do NS-3 proposto para avaliar algumas redes WirelessHART em diferentes cenários. Os resultados aqui apresentados foram publicados em [Nobre et al. 2015*a*]. Nosso objetivo principal é demonstrar as características chave do módulo, tais como: suporte às topologias utilizadas no WirelessHART, escalonamento e roteamento, e cálculo do consumo de energia (tempo de vida do dispositivo), confiabilidade, falhas transientes (injeção de ruído), um modelo de erro realístico e flexível (erros individuais por link e erro dependente do tamanho dos pacotes). É importante ressaltar que questões a respeito de atenuação e posicionamento e tratados no artigo anterior [Nobre et al. 2010].

As principais premissas para as simulações neste capítulo são:

- **Cenários de simulação**: Foram utilizadas topologias em linha, estrela e cluster (um caso particular de uma topologia em malha).
- **Atenuação**: se dois ou mais dispositivos são vizinhos, então é assumido que estes estão sempre dentro do alcance de rádio um do outro. Foi adotado modelo de propagação Friss por questão de simpicidade uma vez que é nativo ao NS–3.
- **Mobilidade**: os dispositivos têm posicionamento fixo.
- **Probabilidade de Erro**: As probabilidades de erro utilizadas como parâmetros para o modelo de erro de Gilbert/Elliot são listadas na Tabela 6.1. Quatro cenários foram definidos. O Caso I é um cenário otimista, e a proporção entre T_g e T_b é por volta de 100. A mesmo métrica para os Casos II, III e IV (os mais pessimistas) são respectivamente: 20 ,20 e 8. É importante destacar que quanto maior a proporção, maior o tempo gasto no estado bom do modelo Gilbert/Elliot.
- **Injeção de interferência**: as probabilidades de erro para cada canal em um dado link pode ser configurado individualmente. Estas configurações de probabilidades podem ser escalonadas para ocorrer em um tempo de simulação definido pelo usuário.

- **Roteamento e escalonamento**: uma vez que o WirelessHART não define um algoritmo para roteamento e escalonamento, mas provê uma sugestão de algoritmo, esta sugestão foi adotada nesse experimento.
- **Superframe**: É considerado apenas um único superframe com duração de 10 segundos (definido pelo usuário).
- **Taxa de atualização**: cada dispositivo produz um único pacote de informação por ciclo de superframe.
- **Confiabilidade**: a métrica da confiabilidade é baseada na proporção entre a quantidade de informação produzida por um dispositivo produtor de dados e a quantidade dessa informação que chega ao *gateway*. Dados redundantes não são contabilizados.
- **Estabilidade**: Esta métrica está relacionada a qualidade de um link. É a proporção entre o número de pacotes enviados pela dispositivo de origem do link e número de pacotes recebidos pelo dispositivo de destino do link.
- **Duração de Bateria**: foram simuladas baterias de 1200 mAh de capacidade, como descrito em [Casilari et al. 2010].
- **Tamanho do Pacote**: Foram considerados pacotes industriais típicos de (90 bytes) de acordo com [Silva et al. 2010*a*] e pacotes ACK nativos de 9 bytes.

Tabela 6.1: Probabilidades de erro para comunicação em canais.

Cenário	Pg	Pb	Descrição
Case I	0.9999918	0.999184	Ambiente externo [Wang & Moayeri 1995]
Case II	0.9999	0.998	Cenário de erro em rajada [Willig et al. 2002]
Case III	0.999	0.98	Cenário de erro em rajada [Bhagwat et al. 1997]
Case IV	0.995	0.96	Ambiente Interno [Fantacci & Scardi 1996]

Por uma questão de confiabilidade da simulação, foi obtido um nível de confiança de 99,8% com intervalo de confiança de 1% ou 20.000 de tempo de simulação (para todas as simulações realizadas).

6.1 Topologia estrela

O primeiro cenário simulado foi a topologia em estrela. Considere uma aplicação de monitoramento da temperatura em quatro caldeiras. Um dispositivo de campo é instalado em cada caldeira como descrito na Figura 6.1. A informação produzida pelos diferentes

nós produtores retratados em cinza e as rotas de dados representadas pelas setas. Todos os dados produzidos pelos nós produtores são direcionados ao *gateway*. A estratégia de redundância utilizada é a de que se um link falha na primeira transmissão, uma retransmissão é realizada nó próximo *slot* de tempo reservado para aquela estação. Caso a retransmissão falhe, o pacote é descartado. Note que se a primeira transmissão for bem sucedida mas o pacote de confirmação for perdido, a retransmissão deverá ser utilizada.

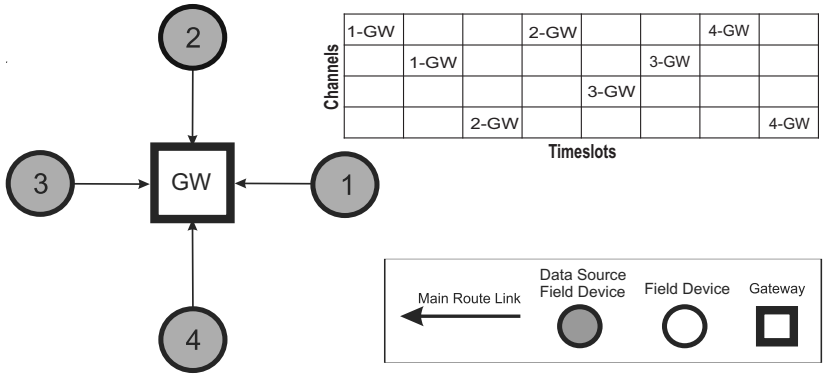


Figura 6.1: Topologia estrela

Para este experimento cada link foi configurado com uma probabilidade de erro como descrito na Tabela 6.2) de modo a se demonstrar a capacidade do módulo em configurar probabilidades de erro diferentes para cada link.

Tabela 6.2: Valores da estabilidade do link, confiabilidade e tempo de vida para a simulação da topologia estrela

Erro(link)	**Estabilidade do Link**	**Confiabilidade**	**Tempo de vida (Dias)**	**Tempo de vida (Proporção)**
Caso I (1-GW)	98.06%	100.00%	2423	-
Caso II (2-GW)	87.73%	98.15%	2148	11.34%
Caso III (3-GW)	40.74%	64.70%	1567	35.32%
Caso IV (4-GW)	1.35%	2.70%	1358	43.95%

De acordo com a Tabela 6.2, quanto mais pessimista é o modelo, maior é quantidade de energia consumida, reduzindo o tempo de vida do dispositivo. Comparando os *links* 4-GW (modelo de erro – Caso IV) e 1-GW (modelo de erro - Caso I), é possível notar que o primeiro consome 44% de energia a mais que o segundo. Este comportamento pode ser explicado pelo fato de que dispositivos que se comunicam por ambientes mais ruidosos perdem mais pacotes e consequentemente recorrem mais as retransmissões para

completar a transmissão dos dados. Adicionalmente foi verificado (como esperado) que a confiabilidade para cada dispositivo é ligeiramente maior que a estabilidade. Em uma topologia estrela há apenas hops singulares. Logo, a maneira de se contornar a interferência é transmitir os dados várias vezes. Se a estabilidade do link é ruim, as retransmissões ainda melhoram a confiabilidade dos dispositivos. Por exemplo, a diferença entre a confiabilidade e a estabilidade para o link 2-GW é por volta de 10%. Isso ocorre por causa do mecanismo de retransmissão. Caso tal mecanismo não existisse, a estabilidade seria semelhante à confiabilidade.

Uma das características mais importantes implementadas no módulo de simulação é a capacidade de se injetar interferência em diferentes momentos da simulação. Para demonstrar esta característica, é assumido a mesma topologia em estrela descrita na Figura 6.1, entretanto é utilizado apenas um dispositivo de campo e o *gateway*. A métrica de avaliação é a confiabilidade instantânea, que mede continuamente a confiabilidade para respectivo dispositivo de campo no instante t. *Links* com três cenários de erro foram utilizados: Caso I, Caso III e injeção de interferência. O terceiro cenário de erro é uma mistura do Caso I com o Caso III. Em tal cenário a comunicação é iniciada com o Caso I, contudo, o link é reconfigurado entre 5000 e 9000 segundos para utilizar o Caso III. Os resultados são mostrados na Figura 6.2.

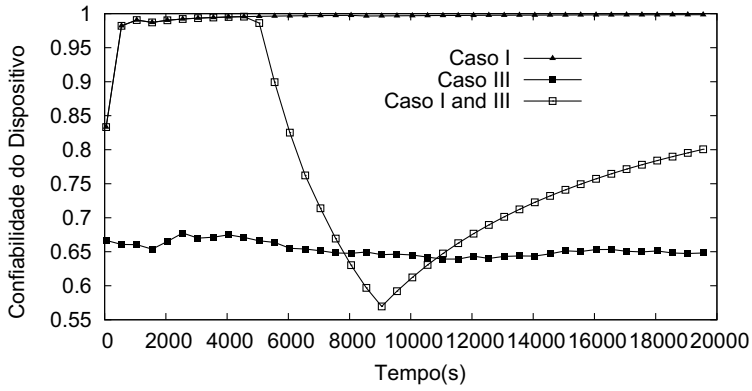


Figura 6.2: Influência da injeção de Interferência na confiabilidade do dispositivo.

Como esperado, a confiabilidade para para o cenário com injeção de interferência declina no instante 5.000 segundos. Em outras palavras, o link de comunicação estava bom (Caso I) até 5000 segundos e em seguida recebeu a influência da interferência (Caso III). Até que em 9000 segundos a interferência cessa (a configuração do link retorna ao Caso I) e a confiabilidade instantânea cresce continuamente.

6.2 Topologia em linha

A topologia em linha é uma solução típica utilizada para aplicações de monitoramento em oleodutos. Nesse caso a informação é encaminhada de dispositivo a dispositivo até que se chegue ao *gateway*. A Figura 6.3 escreve um exemplo de topologia em linha. Para um melhor entendimento, nós assumimos que o último dispositivo de campo é o produtor dos dados. Os outros dispositivos apenas encaminham a informação produzida.

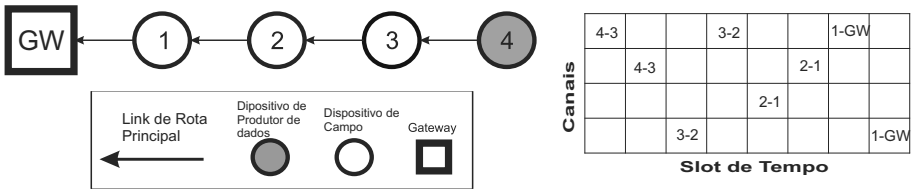


Figura 6.3: Topologia em Linha.

Para a topologia em linha, são conduzidos dois experimentos. O primeiro está relacionado à confiabilidade do dispositivo inicial (o que produz a informação) da topologia em linha. Por outro lado, o segundo experimento analisa a influência do modelo de erro no consumo de energia e na confiabilidade. Um requisito fundamental para qualquer rede é identificar o dispositivo mais crítico. Em uma topologia em linha, o dispositivo inicial é considerado o mais sensível porque este depende de todos os outros dispositivos em sequência para alcançar o *gateway*. Portanto, a confiabilidade do dispositivo inicial foi avaliada, assumindo-se diferentes tamanhos de rede. Os resultados estão descritos na Figura 6.4. Todos os *links* foram configurados com cenário de erro Caso II. Como esperado, a confiabilidade do dispositivo é sensível ao tamanho da rede. A confiabilidade decai quando a rede cresce em tamanho, seguindo uma tendência exponencial. Isso pode ser explicado pelo fato de que com mais saltos, o pacote está mais suscetível a influência de interferências, uma vez que será transmitido pelo meio mais vezes.

O estudo do impacto da interferência sobre os *links* da rede é fundamental para aplicações industriais. Dado um cenário específico, o desenvolvedor da rede pode estimar ações durante os primeiros estados do desenvolvimento para garantir os requisitos de dependabilidade para o sistema. Com isso, o segundo experimento conduzido na topologia em linha temcpor objetivo avaliar tal impacto. É utilizada a topologia descrita na Figura 6.3. Os *links* foram configurados com o Caso I de probabilidade de erro (o mais otimista). Para avaliar o impacto da interferência, quatro subcenários foram adotados: cada um tem um link ruim, de má qualidade (configurado com probabilidade de erro do Caso IV). Os resultados são descritos na Figura 6.5 e na Tabela 6.3.

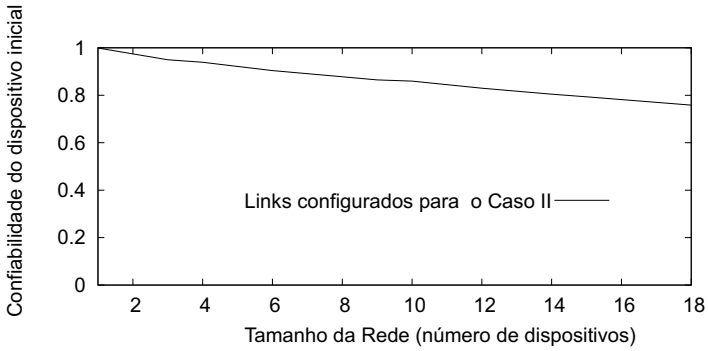


Figura 6.4: Avaliação de confiabilidade para o dispositivo inicial.

De acordo com a Figura 6.5, o transmissor do link ruim sempre tem uma duração de bateria menor que os outros dispositivos da rede. Este comportamento pode ser atribuído ao fato de que estes dispositivos realizam mais retransmissões em ambientes ruidosos e consequentemente consomem mais energia do que outros dispositivos. Este comportamento é visto em uma forma mais branda no receptor do link ruim, uma vez que este recebe mais pacotes (devido as retransmissões) do que os outros dispositivos. No experimento da topologia em linha, o dispositivo de origem tem um menor consumo de energia quando comparado aos outros dispositivos uma vez que este não recebe pacotes de dados, apenas os transmite. O *gateway* tem um comportamento similar, uma vez que este apenas recebe pacotes de dados mas não os envia. Os dispositivos entre o link ruim e o *gateway* tem um consumo menor uma vez que os pacotes raramente passam pelo link ruidoso, fazendo com que tais dispositivos transmitam e recebam menos. Finalmente, os dispositivos entre o dispositivo de origem e o link ruim mantêm o mesmo padrão de consumo de energia. Entretanto, este consumo é maior que o encontrado nos dispositivos após o link ruim. Isto pode ser explicado pelo fato de que os dispositivos entre o dispositivo de origem e o link ruim são responsáveis por encaminhar os pacotes, logo consumindo mais energia. O segundo experimento também avalia a confiabilidade do dispositivo de origem Os resultados são mostrados na Tabela 6.3. *Links* de má qualidade são inseridos alternadamente na topologia até que se alcance o *gateway*. Devido a simetria desta abordagem, foi verificada (como esperado) que a influência do link de má qualidade foi semelhante

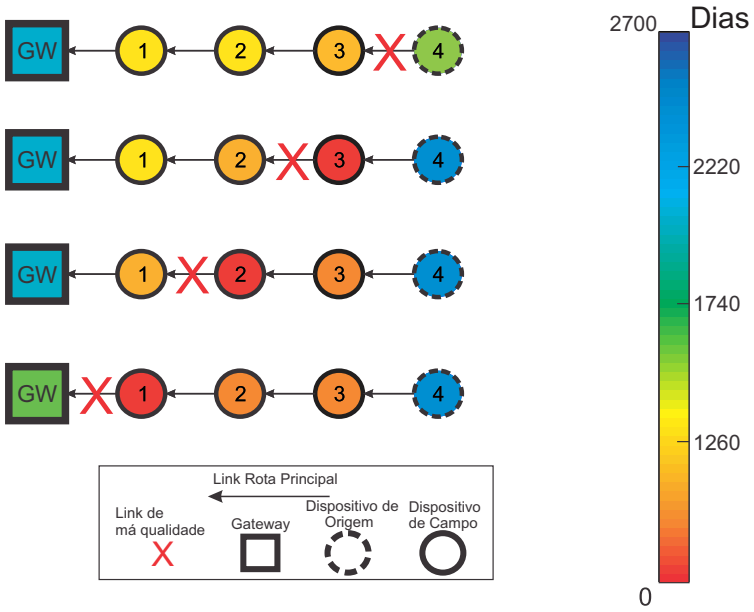


Figura 6.5: Influência de um link de má qualidade no consumo de energia.

nos quatro sub cenários. Note que os resultados apresentados na Tabela 6.3 são considerados os mesmos, uma vez que todos então dentro do intervalo de confiança de 1%. Na Figura 6.4, é verificado que a confiabilidade do dispositivo inicial para a rede com quatro dispositivos de campo é por volta de 94%. Logo, comparando-se este resultado com o descrito na Tabela 6.3, é possível perceber que a injeção de interferência reduzio a confiabilidade em 33%. Este resultado mostra o quão importante é a adoção de um mecanismo para superar a influência da interferência na rede.

Tabela 6.3: Influência do link de má qualidade na confiabilidade do dispositivo inicial.

Link de má qualidade	$Fd_4 \rightarrow Fd_3$	$Fd_3 \rightarrow Fd_2$	$Fd_2 \rightarrow Fd_1$	$Fd_1 \rightarrow GW$
Confiabilidade	63.85%	63.80%	64.10%	64.25%

6.3 Topologia cluster

A topologia cluster é normalmente utilizada quando há a necessidade de segregar parcialmente a rede. Cada cluster deve assumir tarefas específicas, por exemplo monitorar um equipamento específico, priorização de tráfego, prover redundância (novos caminhos), etc. A Figura 6.6 descreve um cluster típico da tecnologia WirelessHART. Para um

melhor entendimento, é assumido que apenas os dispositivos folha são fontes de dados. Os cluster se comunicam se ligam entre si através dos dispositivos roteadores.

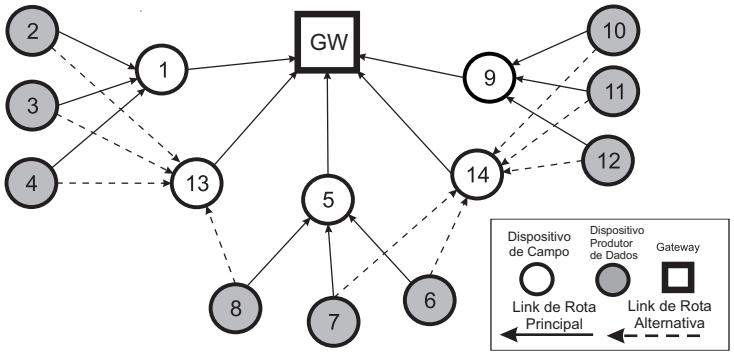


Figura 6.6: Topologia Cluster.

Baseado na topologia cluster, á possível avaliar uma característica importante apresentada na norma WirelessHART: a redundância temporal. De acordo com o padrão, se há uma rota alternativa para o encaminhamento da mensagem, então o dispositivo deve enviar o mesmo pacote pelas rotas alternativa e principal. Dois *slots* de tempo são reservados para a rota principal e uma para a rota alternativa. Se nenhum ACK for recebido pela rota principal (primeiro *slot* de tempo), o pacote é retransmitido na mesma rota (segundo *slot* de tempo). Não há retransmissão de dados em rotas alternativas. Note que as rotas primárias e secundárias estão destacadas na Figura 6.6. Utilizaremos um exemplo para esclarecer a notação descrita na Figura 6.6. Assumindo que os dispositivos 2 e 6 são produtores de informação, é possível gerar o escalonamento correspondente de acordo com a Tabela 6.4. Implementando este escalonamento demonstramos a flexibilidade do módulo proposto em termos de suporte a transmissão simultânea em canais diferentes. Adicionalmente, o uso das rotas primárias e alternativas pode ser notada no escalonamento. Para o dispositivos 2 e 6 temos:

- **Dispositivo de Campo 2**
  - Rota Principal: $D2 \rightarrow D1 \rightarrow GW$
  - Rota Alternativa: $D2 \rightarrow D13 \rightarrow GW$
- **Dispositivo de campo 6**
  - Rota Principal: $D6 \rightarrow D5 \rightarrow GW$
  - Rota Alternativa: $D6 \rightarrow D14 \rightarrow GW$

Os experimentos conduzidos nesta topologia foram projetados para a avaliação do impacto de diferentes padrões de interferência nos mecanismo de redundância de rotas.

Tabela 6.4: Escalonamento para os dispositivos de campo 2 e 6.

Canal	**Slot de tempo**										
	0	1	2	3	4	5	6	7	8	9	10
0	$2 \rightarrow 1$		$2 \rightarrow 13$				$13 \rightarrow 0$			$14 \rightarrow 0$	
1	$6 \rightarrow 5$		$6 \rightarrow 14$					$5 \rightarrow 0$			
2		$6 \rightarrow 5$			$1 \rightarrow 0$						$14 \rightarrow 0$
3		$2 \rightarrow 1$				$13 \rightarrow 0$			$5 \rightarrow 0$		
4				$1 \rightarrow 0$							

As métricas adotadas foram o consumo de energia e a confiabilidade dos dispositivos. A topologia estudada foi dividida em três clusters de acordo com a Figura 6.7. Os *links* dos clusters I, II e III são configurados com probabilidades de erro do Caso I, Caso II e caso III respectivamente. A probabilidade de erro de cada cluster foi aplicada a todos os *links* originados de um dispositivo pertencente ao cluster. Como os clusters são similares, então as mesmas probabilidades de erro provocarão comportamentos semelhantes. Então foi optado pela configuração de três probabilidades de erro de modo a se obter uma variedade de comportamentos no mesmo experimento, tornando assim mais fácil a comparação entre estes comportamentos. Adicionalmente em um cenário real como o de uma refinaria, é possível a existência de diferentes probabilidades de erro ao longo da planta.

Os resultados sobre o consumo de energia são apresentados na Figura 6.7. Em geral, os dispositivos produtores de dados apresentam um consumo menor do que os dispositivos roteadores. Isto pode ser causado pelo fato de que o primeiro apenas transmite pacote de dados mas não os recebe, enquanto os roteadores concentram os fluxos de dados dos outros dispositivos, levando a um consumo mais elevado de energia. Outra observação é a de que o consumo de energia dos dispositivos folhas nos Clusters II e III é maior que o daqueles do Cluster I. A explicação para este fato é são os *links* ruidosos configurados para os Clusters II e III, os quais causam um maior número de retransmissões. Considerando apenas o consumo de energia dentre os dispositivos roteadores, é possível notar que os roteadores 9 e 14 apresentam menor consumo. Apesar de que estes recebem pacotes para rotear, a maioria deles vem com erros devido a qualidade dos *links* configurados no Cluster III e ele não são encaminhados ao *gateway*. Outra comparação interessante ocorre entre os roteadores 1 e 5. O primeiro recebe menos retransmissões, entretanto, ele encaminha mais pacotes ao *gateway* devido a qualidade do link. Um comportamento oposto é atribuído ao roteador 5. Contudo ambos têm consumo de energia similares. O consumo de energia dos roteadores 13 e 14 também foram analisados. Ambos recebem pacotes originados de

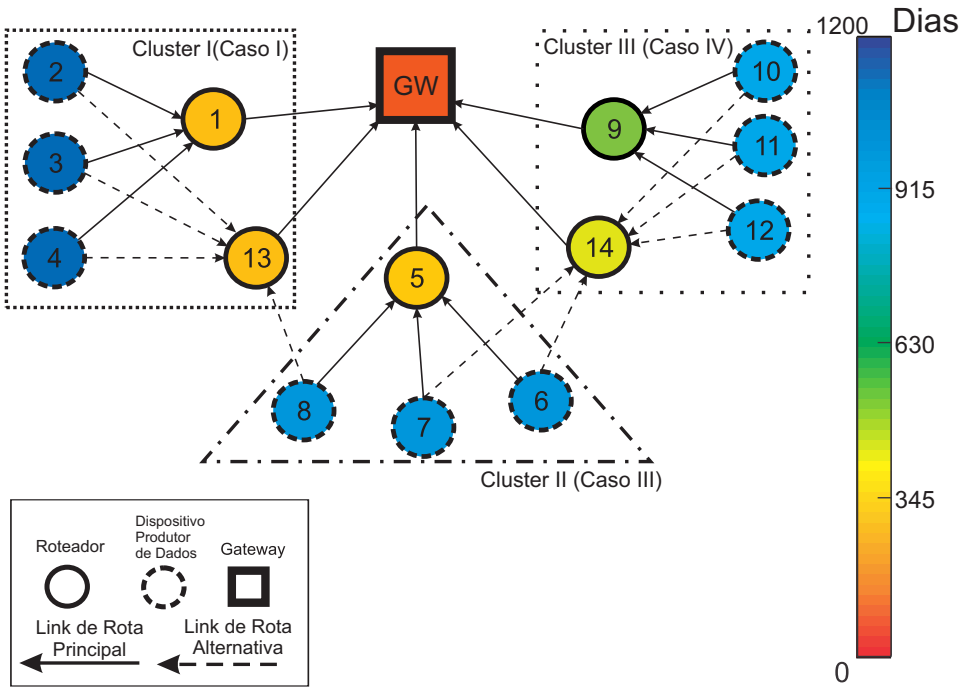


Figura 6.7: Avaliação do consumo de energia para uma topologia cluster com diferentes padrões de interferência.

rotas alternativas. Apesar de possuírem mais *slots* de tempo a disposição o roteador 14 consome menos energia que o roteador 13. Isso se dá, novamente, devido a qualidade dos *links* no Cluster 3. A quantidade de pacotes corrompidos no roteador 14 é maior do que a quantidade no roteador 13, causando o último a transmitir muito mais pacotes. A intenção da segunda parte dos experimentos foi avaliar a confiabilidade dos dispositivos de campo produtores de informação. Os resultados são descritos na Tabela 6.5.

Os produtores de dados no Cluster I apresentaram uma alta confiabilidade apesar do caminho de dois hops até o *gateway*. Este resultado é atribuído ao uso da rota alternativa, o qual eleva a probabilidade de sucesso de transmissão e excelente qualidade do link de transmissão. De acordo com os resultados descritos na topologia estrela (caminho de 1 hop), era esperado que a confiabilidade decairia para um caminho de 2 hops. Este comportamento apenas ocorreu nos Cluster II e III devido as suas pessimistas probabilidades de erro. Um resultado interessante foi observado no roteador 8. Apesar dos *links* ligados a este estarem configurados com uma probabilidade de erro pessimista, este usa uma rota alternativa para o roteador 13. O caminho entre o roteador 13 e o *gateway* utiliza um link muito otimista, com alta taxa de sucesso. Logo a diferença entre resultados para o Cluster II justifica a importância da utilização de uma rota alternativa. O resultado ruim do Cluster

Tabela 6.5: Avaliação de confiabilidade para os dispositivos de campo produtores de informação.

Cluster	Dispositivo produtor	Confiabilidade	Probabilidade de Erro
1	2	100.00%	Caso I
	3	100.00%	Caso I
	4	100.00%	Caso I
2	6	44.30%	Caso III
	7	44.50%	Caso III
	8	64.90%	Caso III
3	10	0.05%	Caso IV
	11	0.10%	Caso IV
	12	0.00%	Caso IV

III ocorre devido a uma combinação de um uma probabilidade de erro pessimista e uma passagem de dois hops para o *gateway*. Note que os resultados do Cluster II demonstram uma melhor performance devido a probabilidade de erro serem menos pessimistas que as do Cluster III.

Capítulo 7

Conclusão e trabalhos futuros

Como mostrado na fundamentação apresentada no Capítulo 2, a tecnologia *Wireless*HART tem uma significativa relevância no atual cenário industrial. Portanto, técnicas que melhorem seu desempenho possuem relevância para o tema.

Tendo em vista a melhoria de desempenho da tecnologia, é importante destacar que o roteamento e o escalonamento utilizados tem influência direta sobre este desempenho. Neste trabalho apresentamos primeiramente um comparativo entre as principais técnicas de escalonamento e roteamento desenvolvidas especificamente para a tecnologia *Wireless*HART. Outras técnicas relevantes para a bibliografia foram abordadas, mas com um enfoque menor. Na análise é notado que o número de soluções cresce com a quantidade de aplicações onde o *Wireless*HART é utilizado. Por exemplo, aplicações de monitoramento podem utilizar redes em malha para melhorar a confiabilidade utilizando redundância espacial (múltiplos caminhos), aplicações de controle podem utilizar redundância temporal (múltiplas cópias dos dados transmitidos). Para outras aplicações também existe a possibilidade de soluções híbridas.

Neste trabalho também foi apresentado o algoritmo de escalonamento Flow. Este algoritmo realiza o escalonamento com base nas restrições apresentadas pela norma *Wireless*HART, dá suporte a múltiplas taxas de atualização e suporte ao uso de rotas alternativas. Além disso, podemos destacar como uma das principais características do algoritmo Flow a possibilidade da escolha e combinação das métricas de escalonamento: grau, janela de tempo e criticalidade, sendo essa última uma métrica nova e desenvolvida nesta tese. Por fim podemos destacar o modo como as transmissões são tratadas. A abstração dos fluxos permite que possamos contornar o problema do desperdício de *slots* do algoritmo de escalonamento Han (apresentado em [Han et al. 2011*b*]), lidando com as repetições de transmissão de superframes com menores taxas de atualização e maneira independente.

Para a validação do protocolo foram realizados testes que avaliaram os aspectos de *de-*

lay, taxa de sucesso de alocação de *links*, taxa de conclusão de fluxo e taxa de ocupação do superframe. Os testes comparam o algoritmo Flow com o algoritmo de escalonamento Han, sendo este último o mais utilizado atualmente em redes *Wireless*HART. Os resultados apresentados mostram uma vantagem do algoritmo Flow proposto na maioria dos casos de teste com exceção de topologia Yang nos quesitos taxa de sucesso de alocação de *links* e taxa de conclusão de fluxo. Esta situação pode ser explicada pelo fato de que na topologia Yang não há utilização de rotas alternativas, logo não ocorrendo a situação de desperdício de *slots* do algoritmo Han (explicada no Capítulo 4) e contribuindo para o melhor desempenho do mesmo sobre o algoritmo Flow.

Tendo em vista estes resultados, é mostrado que o algoritmo Flow é uma alternativa principalmente em redes que necessitem das rotas alternativas e naquelas em que as taxas de atualização dos dispositivos são iguais. Uma vez que o próprio algoritmo Han, nesta segunda situação, recai em uma técnica *First-fit*, uma vez que a sua métrica é baseada apenas na taxa de publicação.

A terceira parte desta tese consiste na apresentação e um simulador de redes *Wireless*HART baseado no simulador de redes NS-3, com o gerador de escalonamento e uma camada física mais realista. Para a validação da proposta, foram conduzidos experimentos (sobre diferentes probabilidades de erro) para avaliar a confiabilidade fim-a-fim, consumo de energia, o comportamento da rede sobre influência de canais ruidosos, o impacto do número de saltos no nível de confiabilidade e os impactos da existência de rotas alternativas. Os testes de validação do escalonamento foram realizados previamente para validação do algoritmo Flow, portanto o escalonamento e o roteamento nestes testes de validação das camadas inferiores foram implementados de acordo com a sugestão da norma *Wireless*HART. A possibilidade de configurar diferentes probabilidades de erro para cada canal de cada *link* em qualquer tempo de simulação, permite que o módulo de simulação avalie cenários mais realistas. Todos os experimentos são passíveis de serem reproduzidos e pode servir como base apara a avaliação de nova tecnologias industriais emergentes.

Como indicações de trabalhos futuros relacionados a revisão mostrada no Capítulo 3, sugerimos estudos das métricas de modo a se incluir novas características (ex.: robustez), a utilização de funções ponderadas dos algoritmos de roteamento na parte do processo de decisão do escalonamento e adaptação automática dos valores dos pesos das funções ponderadas com base no ciclo de vida da rede. Com base no fato de que a maioria dos algoritmos de escalonamento mostrados utilizam *slots* dedicados para o escalonamento, podemos propor também o utilização de *slots* compartilhados no escalonamento (como em [Zhang et al. 2012]) para uma melhoria na confiabilidade. Como outro ponto importante para destacarmos seria a questão dos tráfegos heterogêneos apresentada em [Li

et al. 2014], que não foi ainda completamente pesquisada na literatura, tornando-se relevante como trabalho futuro. Finalmente, um outro possível estudo a ser desenvolvido seria a respeito do impacto no comportamento da rede da concentração das transmissões em regiões específicas do superframe.

Para trabalhos futuros sobre o algoritmo Flow, podemos indicar a adoção de uma função ponderada para as decisões de escalonamento, a realização de experimentos comparativos com o algoritmo de escalonamento Han utilizando a condição de utilização de dois *access points* e a adoção de novas métricas como qualidade do *link* e consumo de energia.

Finalmente, para trabalhos futuros relativos ao módulo de simulação *Wireless*HART, podemos destacar a implementação de algoritmos de roteamento da literatura a sua integração com os algoritmos de escalonamento, novos algoritmos de escalonamento, a integração do simulador com sistemas reais e anexar um terminal para a execução direta do NS-3 quando operando em sistemas Linux que possuam o NS-3 instalado.

Referências bibliográficas

Ahn, Gahng-Seop, Se Gi Hong, Emiliano Miluzzo, Andrew T. Campbell & Francesca Cuomo (2006), Funneling-mac: A localized, sink-oriented mac for boosting fidelity in sensor networks, *em* 'Proceedings of the 4th International Conference on Embedded Networked Sensor Systems', SenSys '06, ACM, New York, NY, USA, pp. 293–306. **URL:** *http://doi.acm.org/10.1145/1182807.1182837*

Akerberg, J., F. Reichenbach & M. Bjorkman (2010), Enabling safety-critical wireless communication using wirelesshart and profisafe, *em* 'Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on', pp. 1 –8.

Akerberg, Johan, Mikael Gidlund, Tomas Lennval, Neanderm Jonas & Mats Bjorkman (2011), 'Efficient integration of secure and safety critical industrial wireless sensors networks', *EURASIP Journal on Wireless Communications and Networking* **2011**(1), 100.

Akerberg, Johan, Mikael Gidlund, Tomas Lennvall, Jonas Neander & Mats Björkman (2010), Integration of wirelesshart networks in distributed control systems using profinet io, *em* '8th IEEE International Conference on Industrial Informatics (INDIN)', pp. 154–159.

Akerberg, Johan, Mikael Gidlund, Tomas Lennvall, Jonas Neander & Mats Bjorkman (2011), 'Efficient integration of secure and safety critical industrial wireless sensor networks', *EURASIP Journal on Wireless Communications and Networking* **2011**(1), 100.
URL: *http://jwcn.eurasipjournals.com/content/2011/1/100*

Bai, H. & M. Atiquzzaman (2003), 'Error modeling schemes for fading channels in wireless communications: A survey', *Communications Surveys Tutorials, IEEE* **5**(2), 2–9.

Bhagwat, Pravin, Partha Bhattacharya, Arvind Krishna & SatishK. Tripathi (1997), 'Using channel state dependent packet scheduling to improve tcpthroughput over

wireless lans', *Wireless Networks* **3**(1), 91–102.
URL: *http://dx.doi.org/10.1023/A%3A1019132612232*

Biswas, Sanjit & Robert Morris (2005), 'Exor: Opportunistic multi-hop routing for wireless networks', *SIGCOMM Comput. Commun. Rev.* **35**(4), 133–144.
URL: *http://doi.acm.org/10.1145/1090191.1080108*

Boyes, W. (2011), 'All quiet on the wireless front', *Control (Chicago, Ill)* **24**(8), 28–39.
URL: *http://www.scopus.com/inward/record.url?eid=2-s2.0-80052019064&partnerID=40&md5=23b1c6776f49278a54a89530ee0843f2*

Casilari, Eduardo, Jose M. Cano-García & Gonzalo Campos-Garrido (2010), 'Modeling of Current Consumption in 802.15.4/ZigBee Sensor Motes', *Sensors* **10**(6), 5443–5468.
URL: *http://www.mdpi.com/1424-8220/10/6/5443*

Chen, Deji, Mark Nixon & Aloysius Mok (2010), *WirelessHART Real-Time Mesh Network for Industrial Automation*, Springer.

Cheng, Hongju, Zhihuang Su, Jaime Lloret & Guolong Chen (2014), 'Service-oriented node scheduling scheme for wireless sensor networks using markov random field model', *Sensors* **14**(11), 20940–20962.
URL: *http://www.mdpi.com/1424-8220/14/11/20940*

Colpo, J & D. Mols (2011), 'No strings attached', *Hydrocarbon Engineering* **11**(16), 47–52.

Contiki (2015), 'The Contiki-OS website. Available online: http://www.nsnam.org/ (accessed on 22 January 2015)'.
URL: *http://www.contiki-os.org/start.html*

Cormen, T.H., C. E. Leiserson, R. L. Rivest & C. Stein (2005), *Introduction to Algorithms*, 3ª edição, MIT press.

Costa, Daniel & Luiz Guedes (2010), 'The coverage problem in video-based wireless sensor networks: a survey', *Sensors* **10**, 8215–8247.

Costa, D.G., I. Silva, L.A. Guedes, P. Portugal & F. Vasques (2014), Selecting redundant nodes when addressing availability in wireless visual sensor networks, *em* 'Industrial Informatics (INDIN), 2014 12th IEEE International Conference on', pp. 130–135.

Coutinho, Mauro Margalho (2003), *NETWORK SIMULATOR Guia Básico para Iniciantes*.

Dang, Kui, Ji-Zhong Shen, Li-Da Dong & Yong-Xiang Xia (2013), 'A graph route-based superframe scheduling scheme in wirelesshart mesh networks for high robustness', *Wireless personal communications* **71**(4), 2431–2444.

Dawson-Haggerty, S., A. Tavakoli & D. Culler (2010), Hydro: A hybrid routing protocol for low-power and lossy networks, *em* 'Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on', pp. 268–273.

De Couto, Douglas S. J., Daniel Aguayo, John Bicket & Robert Morris (2003), A high-throughput path metric for multi-hop wireless routing, *em* 'Proceedings of the 9th Annual International Conference on Mobile Computing and Networking', MobiCom '03, ACM, New York, NY, USA, pp. 134–146.
URL: *http://doi.acm.org/10.1145/938985.939000*

Demarch, Douglas (2007), Uma Proposta de Escalonamento Confiável para Redes Sem Fio Baseadas no Padrão IEEE 802.11/11e, Dissertação de mestrado, Universidade Federal de Santa Catarina.

Dickow, Victor Hugo (2014), Avaliacao de Algoritmos de Roteamento e Escalonamento de Mensagens para Redes WirelessHART (In portuguese), Dissertação de mestrado, Universidade Federal do Rio Grande do Sul.

Djukic, P. & S. Valaee (2009), 'Delay aware link scheduling for multi-hop tdma wireless networks', *Networking, IEEE/ACM Transactions on* **17**(3), 870–883.

Ergen, Sinem Coleri & Pravin Varaiya (2010), 'Tdma scheduling algorithms for wireless sensor networks', *Wireless Networks* **16**(4), 985–997.
URL: *http://dx.doi.org/10.1007/s11276-009-0183-0*

Fantacci, R. & M. Scardi (1996), 'Performance evaluation of preemptive polling schemes and arq techniques for indoor wireless networks', *Vehicular Technology, IEEE Transactions on* **45**(2), 248–257.

Felser, M. & T. Sauter (2002), The fieldbus war: history or short break between battles?, *em* 'Factory Communication Systems, 2002. 4th IEEE International Workshop on', pp. 73–80.

Ganesan, Deepak, Ramesh Govindan, Scott Shenker & Deborah Estrin (2001), 'Highly-resilient, energy-efficient multipath routing in wireless sensor networks', *SIGMOBILE Mob. Comput. Commun. Rev.* **5**(4), 11–25.
URL: *http://doi.acm.org/10.1145/509506.509514*

Gao, Guangchao, Heng Zhang & Li Li (2012), An opnet-based simulation approach for the deployment of wirelesshart, *em* 'Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on', pp. 2120–2124.

Gao, Guangchao, Heng Zhang & Li Li (2013), 'A reliable multipath routing strategy for wirelesshart mesh networks using subgraph routing?', *Journal of Computational Information Systems* **9**(5), 2001–2008.

Gungor, V. C. & F. C. Lambert (2006), 'A survey on communication networks for electric system automation', *Computer Networks: The International Journal of Computer and Telecommunications Networking* **50**(7), 877–897.

Gungor, V.C. & G.P. Hancke (2009), 'Industrial wireless sensor networks: Challenges, design principles, and technical approaches', *Industrial Electronics, IEEE Transactions on* **56**(10), 4258–4265.

Han, Bo & Seungjoon Lee (2007), Efficient Packet Error Rate Estimation in Wireless Networks, *em* 'Testbeds and Research Infrastructure for the Development of Networks and Communities, 2007. TridentCom 2007. 3rd International Conference on', pp. 1–9.

Han, Song, Xiuming Zhu, A.K. Mok, Deji Chen & M. Nixon (2011*a*), Reliable and real-time communication in industrial wireless mesh networks, *em* 'Real-Time and Embedded Technology and Applications Symposium (RTAS), 2011 17th IEEE', pp. 3–12.

Han, Song, Xiuming Zhu, A.K. Mok, Deji Chen & M. Nixon (2011*b*), Reliable and real-time communication in industrial wireless mesh networks, *em* 'Real-Time and Embedded Technology and Applications Symposium (RTAS), 2011 17th IEEE', pp. 3–12.

Hao, Jia, Jiechang Wu & Chaoyou Guo (2011), Modeling and simulation of can network based on opnet, *em* 'Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on', pp. 577–581.

Ibrahim, A., Zhu Han & K.J.R. Liu (2008), 'Distributed energy-efficient cooperative routing in wireless networks', *Wireless Communications, IEEE Transactions on* **7**(10), 3930–3941.

(IEC), International Electrotechnical Commission Committee (2010), *IEC 62591: Industrial communication networks—Wireless communication network and communications profiles—WirelessHART*.

(IEC), International Electrotechnical Commission Committee (2012), *IEC 62734: Industrial communication networks — Fieldbus specifications —Wireless systems for industrial automation: process control and related applications (based on ISA100.11a)*.

IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs) (2006), *IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)* pp. 1–320.

Jain, Kamal, Jitendra Padhye, Venkata N. Padmanabhan & Lili Qiu (2003), Impact of interference on multi-hop wireless network performance, *em* 'Proceedings of the 9th Annual International Conference on Mobile Computing and Networking', MobiCom '03, ACM, New York, NY, USA, pp. 66–80.
URL: *http://doi.acm.org/10.1145/938985.938993*

Jindong, Zhao, Liang Zhenjun & Zhao Yaopei (2009), Elhfr: A graph routing in industrial wireless mesh network, *em* 'Information and Automation, 2009. ICIA '09. International Conference on', pp. 106–110.

Kai-Di Chang, Jiann-Liang Chen (2012), 'A Survey of Trust Management in WSNs, Internet of Things and Future Internet', *KSII Transactions on Internet and Information Systems(TIIS)* **6**, 5–23.
URL: *http://www.dbpia.co.kr/Article/1618728*

Konovalov, Igor (2010), A framework for wirelesshart simulations, Technical Report T2010:06, Swedish Institute of Computer Science, Box 1263, SE-164 29 Kista, Sweden.

Kostadinovic, M., M. Stojcev, Z. Bundalo & D. Bundalo (2009), Design, implementation and simulation of wirelesshart network, *em* 'Telecommunication in Modern

Satellite, Cable, and Broadcasting Services, 2009. TELSIKS '09. 9th International Conference on', pp. 556–559.

Krishnamachari, B., D. Estrin & S. Wicker (2002), The impact of data aggregation in wireless sensor networks, *em* 'Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on', pp. 575–578.

Kumar, S. & S. Chauhan (2011), 'A survey on scheduling algorithms for wireless sensor networks', *International Journal of Computer Applications* **20**.

Lee, Sung-Ju & M. Gerla (2001), Split multipath routing with maximally disjoint paths in ad hoc networks, *em* 'Communications, 2001. ICC 2001. IEEE International Conference on', Vol. 10, pp. 3201–3205 vol.10.

Lessard, A. & M. Gerla (1988), 'Wireless communications in the automated factory environment', *Network, IEEE* **2**(3), 64–69.

Li, Liang, R.G. Maunder, B.M. Al-Hashimi & L. Hanzo (2010), 'An energy-efficient error correction scheme for ieee 802.15.4 wireless sensor networks', *Circuits and Systems II: Express Briefs, IEEE Transactions on* **57**(3), 233 –237.

Li, Yan, Haibo Zhang, Zhiyi Huang & M. Albert (2014), Optimal link scheduling for delay-constrained periodic traffic over unreliable wireless links, *em* 'INFOCOM, 2014 Proceedings IEEE', pp. 1465–1473.

Liang, Wei, Xiaoling Zhang, Yang Xiao, Fuqiang Wang, Peng Zeng & Haibin Yu (2011), 'Survey and experiments of wia-pa specification of industrial wireless network', *Wireless Communications and Mobile Computing* **11**(8), 1197–1212.
URL: *http://dx.doi.org/10.1002/wcm.976*

Lin, Ying-Dar, Chun-Nan Lu, Yuan-Cheng Lai, Wei-Hao Peng & Po-Ching Lin (2009), 'Application classification using packet size distribution and port association', *Journal of Network and Computer Applications* **32**(5), 1023 – 1030. Next Generation Content Networks.
URL: *http://www.sciencedirect.com/science/article/pii/S1084804509000484*

Macedo, D., L.A. Guedes & I. Silva (2014), A dependability evaluation for internet of things incorporating redundancy aspects, *em* 'Networking, Sensing and Control (ICNSC), 2014 IEEE 11th International Conference on', pp. 417–422.

Manges, W., J.N. Sorge & C.W. Taft (2012), Industrial wireless sensors: A user's perspective on the impact of standards on wide-spread deployment, Vol. 492, pp. 195–203.

Marina, M.K. & S.R. Das (2001), On-demand multipath distance vector routing in ad hoc networks, *em* 'Network Protocols, 2001. Ninth International Conference on', pp. 14–23.

Mathworks (2015), 'MathWorks-MATLAB and Simulink for Technical Computing website. Available online: http://www.mathworks.com/ (accessed on 22 January 2015)'. **URL:** *http://www.mathworks.com*

Mueller, Stephen, Rosep. Tsang & Dipak Ghosal (2004), Multipath routing in mobile ad hoc networks: Issues and challenges, *em* 'IN PERFORMANCE TOOLS AND APPLICATIONS TO NETWORKED SYSTEMS, VOLUME 2965 OF LNCS', Springer-Verlag, pp. 209–234.

Nhon, Tran & Dong-Seong Kim (2015), 'Real-time message scheduling for isa100.11a networks', *Computer Standards and Interfaces* **37**(0), 73 – 79. **URL:** *http://www.sciencedirect.com/science/article/pii/S0920548914000762*

Nixon, Mark (2012), A comparison of wirelesshart and isa100.11a, Relatório Técnico MSU-CSE-06-2, Emerson Process Management, Round Rock, TX, USA.

Nobre, M., I. Silva & L.A. Guedes (2014), Reliability evaluation of wirelesshart under faulty link scenarios, *em* 'Industrial Informatics (INDIN), 2014 12th IEEE International Conference on', pp. 676–682.

Nobre, M., I. Silva, L.F. Guedes & P. Portugal (2010), Towards a WirelessHART module for the ns-3 simulator, *em* 'Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on', pp. 1–4.

Nobre, Marcelo, Ivanovitch Silva & Luiz Affonso Guedes (2015*a*), 'Performance evaluation of wirelesshart networks using a new network simulator 3 module', *Computers and Electrical Engineering* (0), –. **URL:** *http://www.sciencedirect.com/science/article/pii/S0045790614001311*

Nobre, Marcelo, Ivanovitch Silva & Luiz Affonso Guedes (2015*b*), 'Routing and scheduling algorithms for wirelesshartnetworks: A survey', *Sensors* **15**(5), 9703–9740. **URL:** *http://www.mdpi.com/1424-8220/15/5/9703*

Noh, Donggeon, Ikjune Yoon & Heonshik Shin (2008), 'Low-latency geographic routing for asynchronous energy-harvesting wsns', *Journal of Networks* **3**(1).

NS-3 Consortium (2015), 'The Network Simulator 3 (NS-3) website. Available online: http://www.nsnam.org/ (accessed on 22 January 2015) '.
URL: *http://www.nsnam.org*

Osterlind, F., A. Dunkels, J. Eriksson, N. Finne & T. Voigt (2006), Cross-level sensor network simulation with cooja, *em* 'Local Computer Networks, Proceedings 2006 31st IEEE Conference on', pp. 641–648.

Osterlind, Fredrik (2006), A sensor network simulator for the contiki os, Technical Report T2006:05, Swedish Institute of Computer Science.

Petersen, S. & S. Carlsen (2011), 'Wirelesshart versus isa100.11a: The format war hits the factory floor', *Industrial Electronics Magazine, IEEE* **5**(4), 23–34.

Pham, Tung-Linh & Dong-Seong Kim (2013), Lossy link-aware routing algorithm for isa100.11a wireless networks, *em* 'Industrial Informatics (INDIN), 2013 11th IEEE International Conference on', pp. 624–629.

Pham, Tung-Linh & Dong-Seong Kim (2014), 'Routing protocol over lossy links for isa100.11a industrial wireless networks', *Wireless Networks* **20**(8), 2359–2370.
URL: *http://dx.doi.org/10.1007/s11276-014-0747-5*

Project, NS-3 (2010), 'NS-3 Tutorial and Manual'.
URL: *http://www.nsnam.org/tutorials.html*

Quang, Pham Tran Anh & Dong-Seong Kim (2014), 'Throughput-aware routing for industrial sensor networks: Application to isa100.11a', *Industrial Informatics, IEEE Transactions on* **10**(1), 351–363.

Ramachandran, Iyappan, Arindam K. Das & Sumit Roy (2007), 'Analysis of the contention access period of IEEE 802.15.4 MAC', *ACM Transactions on Sensor Networks (TOSN)* **3**(4), 4.

Ramanathan, S. (1999), 'A unified framework and algorithm for channel assignment in wireless networks', *Wireless Networks* **5**(2), 81–94.
URL: *http://dx.doi.org/10.1023/A%3A1019126406181*

Rhee, Injong, A. Warrier, M. Aia, Jeongki Min & M.L. Sichitiu (2008), 'Z-mac: A hybrid mac for wireless sensor networks', *Networking, IEEE/ACM Transactions on* **16**(3), 511–524.

Riverbead Technology (2015), 'The Riverbed website. Available online: http://www.riverbed.com/ (accessed on 22 January 2015)'.
URL: *http://www.riverbed.com*

Saaty, T.L. (1980), 'The analytic hierarchy process', *Wireless Communications, IEEE Transactions on* **7**(10), 3930–3941.

Saifullah, A., You Xu, Chenyang Lu & Yixin Chen (2010), Real-time scheduling for wirelesshart networks, *em* 'Real-Time Systems Symposium (RTSS), 2010 IEEE 31st', pp. 150–159.

Saifullah, A., You Xu, Chenyang Lu & Yixin Chen (2011), Priority assignment for real-time flows in wirelesshart networks, *em* 'Real-Time Systems (ECRTS), 2011 23rd Euromicro Conference on', pp. 35–44.

Sauter, T. (2010), 'The three generations of field-level networks x2014;evolution and compatibility issues', *Industrial Electronics, IEEE Transactions on* **57**(11), 3585–3595.

Sauter, T., S. Soucek, W. Kastner & D. Dietrich (2011), 'The evolution of factory and building automation', *Industrial Electronics Magazine, IEEE* **5**(3), 35–48.

Sauter, Thilo (2005), *Linking factory floor and the internet*, 2ª edição, CRC Press.

Schutz, H.A. (1988), 'The role of map in factory integration', *Industrial Electronics, IEEE Transactions on* **35**(1), 6–12.

Shah, K., T. Seceleanu & M. Gidlund (2010), Design and implementation of a wirelesshart simulator for process control, *em* 'Industrial Embedded Systems (SIES), 2010 International Symposium on', pp. 221–224.

Shen, Wei, Tingting Zhang, F. Barac & M. Gidlund (2014), 'Prioritymac: A priority-enhanced mac protocol for critical traffic in industrial wireless sensor and actuator networks', *Industrial Informatics, IEEE Transactions on* **10**(1), 824–835.

Shen, Wei, Tingting Zhang, Mikael Gidlund & Felix Dobslaw (2013), 'Sas-tdma: a source aware scheduling algorithm for real-time communication in industrial wireless sensor networks', *Wireless Networks* **19**(6), 1155–1170.
URL: *http://dx.doi.org/10.1007/s11276-012-0524-2*

Silva, I., L.A. Guedes & F. Vasques (2008), Performance evaluation of a compression algorithm for wireless sensor networks in monitoring applications, *em* 'Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on', pp. 672–678.

Silva, I., L.A. Guedes & F. Vasques (2010*a*), A new AODV-based routing protocol adequate for monitoring applications in oil amp; gas production environments, *em* 'Factory Communication Systems (WFCS), 2010 8th IEEE International Workshop on', pp. 283–292.

Silva, I., L.A. Guedes & F. Vasques (2010*b*), A new aodv-based routing protocol adequate for monitoring applications in oil amp; gas production environments, *em* 'Factory Communication Systems (WFCS), 2010 8th IEEE International Workshop on', pp. 283–292.

Silva, I., L.A. Guedes, P. Portugal & F. Vasques (2012*a*), Dependability evaluation of wirelesshart best practices, *em* 'Emerging Technologies Factory Automation (ETFA), 2012 IEEE 17th Conference on', pp. 1–9.

Silva, I., L.A. Guedes, P. Portugal & F. Vasques (2012*b*), 'Reliability and availability evaluation of wireless sensor networks for industrial applications', *Sensors* **12**(1), 806–838.
URL: *http://www.mdpi.com/1424-8220/12/1/806*

Song, Jianping, Song Han, Al Mok, Deji Chen, Mike Lucas, Mark Nixon & Wally Pratt (2008), WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control, *em* 'Proceedings of the 2008 IEEE Real-Time and Embedded Technology and Applications Symposium', RTAS '08, IEEE Computer Society, Washington, DC, USA, pp. 377–386.
URL: *http://dx.doi.org/10.1109/RTAS.2008.15*

Tarique, Mohammed, Kemal E. Tepe, Sasan Adibi & Shervin Erfani (2009), 'Survey of multipath routing protocols for mobile ad hoc networks.', *J. Network and Computer Applications* **32**(6), 1125–1143.
URL: *http://dblp.uni-trier.de/db/journals/jnca/jnca32.htmlTariqueTAE09*

Thomesse, J.-P. (2005), 'Fieldbus technology in industrial automation', *Proceedings of the IEEE* **93**(6), 1073–1101.

USC, University of Southern California (2015), 'The Network Simulator 2 (NS-2) website. Available online: http://nsnam.isi.edu/nsnam/ (accessed on 22 January 2015)'.
URL: *http://nsnam.isi.edu/nsnam/index.php/User*$_{I}$*nformation*

Vuran, M.C. & I.F. Akyildiz (2009), 'Error control in wireless sensor networks: A cross layer analysis', *Networking, IEEE/ACM Transactions on* **17**(4), 1186 –1199.

Wang, Gengyun (2011), Comparison and Evaluation of Industrial Wireless Sensor Network Standards ISA100.11a and WirelessHART, Dissertação de mestrado, Chalmers University of Technology, Sweden.

Wang, H.S. & N. Moayeri (1995), 'Finite-state markov channel—a useful model for radio communication channels', *IEEE Transactions on Vehicular Technology* **44**(1), 163–171.
URL: *http://www.scopus.com/inward/record.url?eid=2-s2.0-0029246979partnerID=40md5=2b1136592cd198987ebe63a9eb1759e9*

Wang, Yipeng & F. Barac (2013), Implementation of the wirelesshart mac layer in the opnet simulator, *em* 'Computer Science and Network Technology (ICCSNT), 2013 3rd International Conference on', pp. 663–668.

Weingärtner, Elias, Hendrik vom Lehn & Klaus Wehrle (2009), A performance comparison of recent network simulators, *em* 'Proceedings of the IEEE International Conference on Communications 2009 (ICC 2009)', IEEE, Dresden, Germany, pp. 1–5.

Willig, A., M. Kubisch, C. Hoene & A. Wolisz (2002), 'Measurements of a wireless link in an industrial environment using an ieee 802.11-compliant physical layer', *Industrial Electronics, IEEE Transactions on* **49**(6), 1265–1282.

Willig, Andreas & Jean-Pierre Ebert (1999), A Gilbert-Elliot Bit Error Model and the Efficient Use in Packet Level Simulation, Relatório Técnico TKN-99-002, Technical University Berlin.

Woo, Alec, Terence Tong & David Culler (2003), Taming the underlying challenges of reliable multihop routing in sensor networks, *em* 'Proceedings of the 1st International Conference on Embedded Networked Sensor Systems', SenSys '03, ACM, New York, NY, USA, pp. 14–27.
URL: *http://doi.acm.org/10.1145/958491.958494*

Yan, Mao, Kam-Yiu Lam, Song Han, Edward Chan, Qingchun Chen, Pingzhi Fan, Deji Chen & Mark Nixon (2014), 'Hypergraph-based data link layer scheduling for reliable packet delivery in wireless sensing and control networks with end-to-end delay constraints', *Information Sciences* **278**(0), 34 – 55.
URL: *http://www.sciencedirect.com/science/article/pii/S0020025514001030*

Yang, Dong, Youzhi Xu, Hongchao Wang, Tao Zheng, Hao Zhang, Hongke Zhang & M. Gidlund (2015), 'Assignment of segmented slots enabling reliable real-time transmission in industrial wireless sensor networks', *Industrial Electronics, IEEE Transactions on* **62**(6), 3966–3977.

Ye, Z., S.V. Krishnamurthy & S.K. Tripathi (2003), A framework for reliable routing in mobile ad hoc networks, *em* 'INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies', Vol. 1, pp. 270–280 vol.1.

Yu, Kan, M. Gidlund, J. Akerbergy & M. Bjorkman (2013), Low jitter scheduling for industrial wireless sensor and actuator networks, *em* 'Industrial Electronics Society, IECON 2013 - 39th Annual Conference of the IEEE', pp. 5594–5599.

Yu, Kan, Zhibo Pang, Mikael Gidlund, Johan Åkerberg & Mats Björkman (2014), 'Realflow: Reliable real-time flooding-based routing protocol for industrial wireless sensor networks', *International Journal of Distributed Sensor Networks* **2014**.

Zand, P., S. Chatterjea, J. Ketema & P. J. M. Havinga (2011), D-sar: A distributed scheduling algorithm for real-time, closed-loop control in industrial wireless sensor and actuator networks, Technical Report TR-CTIT-11-09, Centre for Telematics and Information Technology University of Twente, Enschede.

Zand, Pouria, Arta Dilo & Paul Havinga (2012), Implementation of wirelesshart in ns-2 simulator, *em* 'Emerging Technologies Factory Automation (ETFA), 2012 IEEE 17th Conference on', pp. 1–8.

Zand, Pouria, Arta Dilo & Paul Havinga (2013), 'D-msr: A distributed network management scheme for real-time monitoring and process control applications in wireless industrial automation', *Sensors* **13**(7), 8239–8284.
URL: *http://www.mdpi.com/1424-8220/13/7/8239*

Zhang, Haibo, M. Albert & A. Willig (2012), Combining tdma with slotted aloha for delay constrained traffic over lossy links, *em* 'Control Automation Robotics Vision (ICARCV), 2012 12th International Conference on', pp. 701–706.

Zhang, Haibo, P. Soldati & M. Johansson (2009), Optimal link scheduling and channel assignment for convergecast in linear wirelesshart networks, *em* 'Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, 2009. WiOPT 2009. 7th International Symposium on', pp. 1–8.

Zhang, Haibo, P. Soldati & M. Johansson (2011), Time- and channel-efficient link scheduling for convergecast in wirelesshart networks, *em* 'Communication Technology (ICCT), 2011 IEEE 13th International Conference on', pp. 99–103.

Zhang, Qun, Feng Li, Lei Ju, Zhiping Jia & Zhaopeng Zhang (2014), Reliable and energy efficient routing algorithm for wirelesshart, *em* X.-h.Sun, W.Qu, I.Stojmenovic, W.Zhou, Z.Li, H.Guo, G.Min, T.Yang, Y.Wu & L.Liu, eds., 'Algorithms and Architectures for Parallel Processing', Vol. 8630 de *Lecture Notes in Computer Science*, Springer International Publishing, pp. 192–203.
URL: *http://dx.doi.org/10.1007/978-3-319-11197-1*$_1$5

Zhang, S., A. Yan & T. Ma (2013), Energy-balanced routing for maximizing network lifetime in wirelesshart, *em* 'International Journal of Distributed Sensor Networks', Vol. 2013, p. 7.

Zhang, Sheng, Guoyong Zhang, Ao Yan, Zhongsheng Xiang & Tianming Ma (2013), A highly reliable link scheduling strategy for wirelesshart networks, *em* 'Advanced Technologies for Communications (ATC), 2013 International Conference on', pp. 39–43.

Zhu, X., T. Lin, S. Han, A. Mok, D. Chen, M. Nixon & E. Rotvold (2012), Measuring wirelesshart against wired fieldbus for control, pp. 270–275.
URL: *http://www.scopus.com/inward/record.url?eid=2-s2.0-84868252841&partnerID=40&md5=cfa3bc21f7186845b1eca9aafd5effd8*