

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

**GUSTAVO PEDROSO CAINELLI**

**PROPOSTA DE ALGORITMO DE  
ESCALONAMENTO DE MENSAGENS PARA  
REDES SEM FIO APLICADAS À  
AUTOMAÇÃO DE FÁBRICA**

Porto Alegre  
2020

**GUSTAVO PEDROSO CAINELLI**

**PROPOSTA DE ALGORITMO DE  
ESCALONAMENTO DE MENSAGENS PARA  
REDES SEM FIO APLICADAS À  
AUTOMAÇÃO DE FÁBRICA**

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

ORIENTADOR: Prof. Dr. Carlos Eduardo Pereira

CO-ORIENTADOR: Prof. Dr. Ivan Müller

Porto Alegre  
2020

**GUSTAVO PEDROSO CAINELLI**

**PROPOSTA DE ALGORITMO DE  
ESCALONAMENTO DE MENSAGENS PARA  
REDES SEM FIO APLICADAS À  
AUTOMAÇÃO DE FÁBRICA**

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

Orientador: \_\_\_\_\_

Prof. Dr. Carlos Eduardo Pereira, UFRGS

Doutor pela Universidade de Stuttgart – Stuttgart, Alemanha

Banca Examinadora:

Prof. Dr. Dennis Brandão, USP

Doutor pela Universidade de São Paulo

Prof. Dr. João César Netto, UFRGS

Doutor pela Universite Catholique de Louvain – Louvain-la-Neuve, Bélgica

Prof. Dr. Edison Pignaton de Freitas, UFRGS

Doutor pela Universidade de Halmstad - Halmstad, Suécia

Coordenador do PPGEE: \_\_\_\_\_

Prof. Dr. João Manoel Gomes da Silva Jr.

Porto Alegre, Março de 2020.

## **AGRADECIMENTOS**

Aos professores, Carlos Eduardo Pereira e Ivan Müller, por proverem a oportunidade de trabalho na área de pesquisa e pela confiança depositada no meu trabalho.

Aos colegas do LASCAR em especial ao Christian Alan Krötz, Filipe Guterres Ferreira, Leomar Mateus Radke, Max Feldman, Rodolfo Hecht, Tiago Rodrigo Cruz e Yuri das Neves Valadão pelo auxílio e incentivo nas tarefas desenvolvidas, e por tornarem o caminho até aqui muito mais agradável.

Aos meus professores do IFRS - Campus Farroupilha, pelo conhecimento transmitido durante a graduação e também pelas recomendações para ingressar no mestrado, em especial ao Gustavo Künzel, meu professor e amigo que me auxiliou durante a graduação e o mestrado.

Aos meus pais e a minha irmã, pelo carinho e cuidado ao longo da minha vida e por me incentivarem desde sempre a estudar e me desenvolver.

A minha noiva Fernanda Padilha Noronha, por todo apoio, incentivo, paciência e carinho durante esta jornada.

Aos meus primos Mônica Cainelli Vedovelli e Douglas da Silva Corso por contribuírem tanto para que eu chegasse até aqui.

Aos amigos e familiares que de alguma forma contribuíram e me apoiaram durante este período.

Aos funcionários do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Rio Grande do Sul, pela assistência durante o período do mestrado.

Ao CNPQ, pela provisão da bolsa de mestrado.

## RESUMO

As redes sem fio industriais são uma alternativa às redes com fio para automação de fábrica. Nesse tipo de rede, geralmente um gerenciador central é responsável por criar e manter a rede de forma segura e eficiente, garantindo confiabilidade e determinismo temporal. Uma das tarefas do gerenciador é propiciar o escalonamento de mensagens, que é reservar os recursos da rede para que os dispositivos possam acessar o meio físico e transmitir seus dados. Para que o gerenciador de rede atenda aos requisitos de automação de fábrica é necessário que o processo de escalonamento seja executado o mais rápido possível. Este trabalho tem como objetivo avaliar o processo de escalonamento de enlaces em redes industriais sem fio, comparando diferentes técnicas e apresentando uma proposta de algoritmo de escalonamento que visa reduzir o tempo necessário para gerar o escalonamento. A técnica proposta realiza um processo de pré-escalonamento reservando recursos previamente. As informações geradas nesta etapa são utilizadas no escalonamento final, reduzindo o tempo de execução do processo. Além disso, é apresentada uma comparação entre a técnica proposta neste trabalho que utiliza múltiplos agrupamento de enlaces (superframes) com o obtido de um gateway que utiliza um único agrupamento de enlaces. Os resultados mostram que a utilização de uma etapa antecessora ao escalonamento reduz o tempo necessário para a execução do processo de escalonamento final, habilitando a rede a operar em aplicações de automação de fábrica. A utilização de múltiplos superframes apresenta vantagens ao método comumente empregado uma vez que propicia economia de recursos, aumentando a escalonabilidade da rede, além de reduzir o consumo de energia dos rádios.

**Palavras-chave:** Escalonamento, Redes sem Fio Industriais, Automação de Fábrica, Tempo de Execução.

## **ABSTRACT**

Industrial wireless networks are an alternative to wired networks for factory automation. In this type of network, a central manager is usually responsible for creating and maintaining the network safely and efficiently, ensuring reliability and temporal determinism. One of the tasks of this central manager is to provide the message scheduling, which is reserving network resources so that devices can access the physical medium and transmit their data. For the network manager to meet the factory automation requirements, it is necessary that the scheduling process is performed as quickly as possible. This work aims to evaluate the link scheduling in industrial wireless networks, comparing different techniques and presenting a proposed scheduling algorithm that aims to reduce the time needed to generate the schedule. The proposed technique performs a process of pre-scheduling by reserving resources in advance. The information generated in this step are used in the final scheduling, reducing the execution time of the process. In addition, a comparison is presented between the proposed technique in this work that uses multiple grouping of links (superframes) with the one obtained from a gateway that uses a single group of links. The results show that using a predecessor step reduces the time necessary for the execution of the final scheduling process, enabling the network to operate in factory automation applications. The use of multiple superframes presents advantages to the commonly used method since provides resource savings, increasing network scalability, besides the fact to reduce radio power consumption.

**Keywords: Link Scheduling, Industrial Wireless Networks, Factory Automation, Multiple Superframes, Execution Time.**

## LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de RSFI. . . . .	13
Figura 2 – Topologia estrela com $i$ dispositivos conectados ao AP. . . . .	14
Figura 3 – Rotina do Gerenciador de Rede. . . . .	16
Figura 4 – Exemplo de aplicação de uma RSFI no controle de movimento de um atuador robótico. . . . .	21
Figura 5 – Superframe no padrão IEEE 802.15.4-2006. . . . .	23
Figura 6 – Topologia estrela. . . . .	28
Figura 7 – Matriz de escalonamento. . . . .	28
Figura 8 – Matriz escalonamento TSCH com e sem reuso frequência. . . . .	29
Figura 9 – Múltiplos superframes na rede. . . . .	31
Figura 10 – Superframes para períodos de 4, 8 e 16 timeslots. . . . .	32
Figura 11 – Superframes e links organizados na matriz de escalonamento. . . . .	32
Figura 12 – Diagrama da arquitetura de rede empregada, de acordo com o protocolo WH. . . . .	40
Figura 13 – Terminal do PC com o gerenciador, o gateway, o NAP-Host e a aplicação. . . . .	40
Figura 14 – Diagrama de classes do gerenciador de rede. . . . .	42
Figura 15 – Dispositivo de campo utilizado nos testes práticos. . . . .	44
Figura 16 – Gateway comercial Emerson 1420A (EMERSON, 2013). . . . .	45
Figura 17 – Aplicação do gateway comercial Emerson 1420A (EMERSON, 2013). . . . .	45
Figura 18 – Sniffer WiAnalys utilizado para captura das mensagens durante os testes práticos. . . . .	46
Figura 19 – Formato do superframe de anúncio e posição dos links. . . . .	49
Figura 20 – Formato do superframe de gerenciamento. . . . .	50
Figura 21 – Formato dos superframes de dados. . . . .	51
Figura 22 – Diagrama de sequência do processo de agregação. . . . .	53
Figura 23 – Diagrama de sequência do processo de solicitação de serviço. . . . .	55
Figura 24 – Divisão do superframe na estratégia de Han. . . . .	59
Figura 25 – Exemplo de escalonamento gerado pelo Han e pelo Han-MO. . . . .	62
Figura 26 – Inserção da etapa de pré-escalonamento na rotina do gerenciador de rede. . . . .	63
Figura 27 – Exemplo de rede com dois dispositivos que publicam a cada 8 <i>timeslots</i> . . . . .	64
Figura 28 – Dispositivos 1 e 2 escalonados no superframe $S_i$ de tamanho 8 . . . . .	64
Figura 29 – Superframes com as estruturas dos períodos 8 e 16 pré-alocadas. . . . .	65
Figura 30 – Conjuntos $C_8$ , $C_{16}$ e $C_{32}$ . . . . .	66
Figura 31 – Fluxograma do algoritmo LSPA. . . . .	69
Figura 32 – Estudo de Caso A - Reescalonamento completo da rede. . . . .	71

Figura 33 – Estudo de Caso B - Escalonamento de um dispositivo agregando-se à rede já formada. . . . .	73
Figura 34 – Processo para gerar vetor de escalonamento para testes do estudo de caso B. . . . .	75
Figura 35 – Exemplo de vetor de escalonamento com diferentes taxas de ocupação. . . . .	76
Figura 36 – Teste realizado com o gerenciador de rede desenvolvido. . . . .	77
Figura 37 – Tempo de execução dos algoritmos de escalonamento considerando todos os dispositivos com mesmo período de publicação. . . . .	79
Figura 38 – Tempo de execução dos algoritmos de escalonamento considerando dois períodos de publicação diferentes entre os dispositivos. . . . .	80
Figura 39 – Tempo de execução dos algoritmos de escalonamento considerando o conjunto de dispositivos com quatro períodos de publicação. . . . .	81
Figura 40 – Taxa de sucesso dos algoritmos em 100 diferentes cenários para cada % de ocupação da matriz M. . . . .	82
Figura 41 – Tempo de execução para cada algoritmo em diferentes % de ocupação da matriz de escalonamento. . . . .	83
Figura 42 – Quantidades de links recebidos pelos dispositivos utilizando um único superframe e utilizando múltiplos superframes. . . . .	87
Figura 43 – Ocupação da banda pelos links de comunicação de um dispositivo com período de publicação de 60 segundos. . . . .	88

## LISTA DE TABELAS

Tabela 1 – Protocolos de comunicação industrial. . . . .	26
Tabela 2 – Principais trabalhos analisados. . . . .	38
Tabela 3 – Comandos enviados pelo dispositivo após escutar o pacote de anúncio. 53	
Tabela 4 – Comandos enviados pelo gerenciador ao dispositivo durante o processo de agregação. . . . .	54
Tabela 5 – Comandos enviados ao gerenciador pelo dispositivo para solicitação de serviço. . . . .	55
Tabela 6 – Conjuntos de estruturas gerados. . . . .	66
Tabela 7 – Matriz de conflitos <i>MC</i> resultante após o processo de escalonamento de quatro dispositivos com períodos 500 ms, 500ms, 1 s e 2 s. . . . .	69
Tabela 8 – Parâmetros do Caso A1. . . . .	72
Tabela 9 – Parâmetros do Caso A2. . . . .	72
Tabela 10 – Parâmetros do Caso A3. . . . .	72
Tabela 11 – Parâmetros do Caso B. . . . .	74
Tabela 12 – Dispositivos utilizados no teste com o gerenciador desenvolvido e com o gerenciador comercial. . . . .	76
Tabela 13 – Lista de links recebidos pelos dispositivos durante o provisionamento pelo gerenciador comercial. . . . .	85
Tabela 14 – Lista de links recebidos pelos dispositivos durante o provisionamento pelo Gerenciador desenvolvido. . . . .	86

## LISTA DE ABREVIATURAS

ACK	<i>Acknowledge</i>
ASN	<i>Absolute Slot Number</i>
DM	<i>Deadline Monotonic</i>
EDF	<i>Earliest Deadline First</i>
FA	Factory Automation
LSPA	<i>Link Scheduling using Pre-Allocated structures</i>
PA	Process Automation
RF	Rádio Frequência
RSFI	Redes sem Fio Industriais
RSL	<i>Received Signal Level</i>
RX	<i>Receive</i>
TX	<i>Transmit</i>
WH	<i>WirelessHART</i>

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	12
1.1	Motivação	15
1.2	Objetivos	18
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	19
2.1	Áreas de aplicação de RSFI	19
2.2	Casos de uso de RSFI em automação de Fábrica.	20
2.3	Protocolos de comunicação industrial	22
2.3.1	IEEE802.15.4	22
2.3.2	WirelessHart, ISA SP100 e WIA-PA	23
2.3.3	WISA/PNO WSAW	24
2.3.4	WirelessHP	24
2.3.5	5G	25
2.4	Escalonamento em Redes sem Fio Industriais	26
<b>3</b>	<b>ANÁLISE DO ESTADO DA ARTE</b>	33
<b>4</b>	<b>MATERIAIS E MÉTODOS</b>	39
4.1	Materiais	39
4.1.1	Gerenciador de rede	40
4.1.2	Gateway	43
4.1.3	NAP-Host e NAP-RCP	43
4.1.4	Dispositivos de campo	44
4.1.5	Gateway Comercial	44
4.1.6	Sniffer	45
4.2	Métodos	46
4.2.1	Processo de Escalonamento	46
4.2.2	Escalonamento com múltiplos superframes	47
4.2.3	Processo de agregação de um dispositivo de campo na rede	51
4.2.4	Algoritmos de escalonamento	56
4.2.5	Proposta de algoritmo de escalonamento	62
<b>5</b>	<b>ESTUDOS DE CASO</b>	70
5.1	Estudo de Caso A: Escalonamento completo	71
5.1.1	Caso A1:	71
5.1.2	Caso A2:	72
5.1.3	Caso A3:	72
5.2	Estudo de Caso B: Escalonamento Parcial	73

<b>5.3</b>	<b>Estudo de Caso C: Múltiplos superframes versus superframe único . . . . .</b>	<b>75</b>
<b>5.4</b>	<b>Métricas . . . . .</b>	<b>77</b>
<b>6</b>	<b>RESULTADOS E ANÁLISE . . . . .</b>	<b>78</b>
<b>6.1</b>	<b>Estudo de Caso A: . . . . .</b>	<b>78</b>
6.1.1	Caso A1 . . . . .	78
6.1.2	Caso A2 . . . . .	79
6.1.3	Caso A3 . . . . .	80
<b>6.2</b>	<b>Estudo de Caso B: . . . . .</b>	<b>81</b>
<b>6.3</b>	<b>Estudo de Caso C: . . . . .</b>	<b>84</b>
<b>7</b>	<b>CONCLUSÕES . . . . .</b>	<b>90</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>92</b>

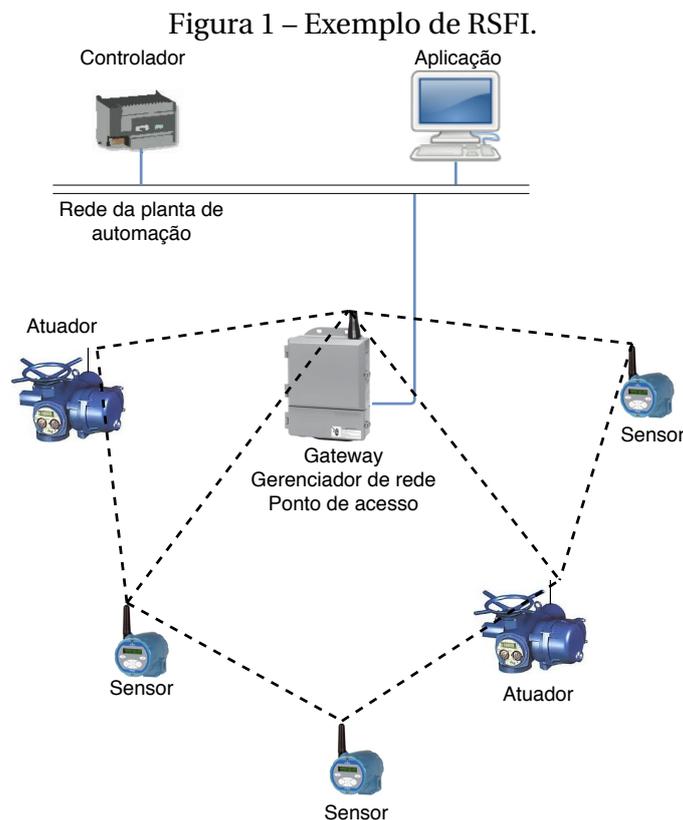
# 1 INTRODUÇÃO

Redes sem Fio Industriais (RSFI) são objeto de estudo por parte da comunidade científica por apresentarem benefícios em relação às redes cabeadas utilizadas atualmente. Algumas das vantagens são a redução de custo de instalação e a mobilidade. No que diz respeito ao custo de instalação, a substituição de cabos por enlaces de rádio reduz significativamente o material utilizado na instalação. Além disso, mesmo que os enlaces de rádios estejam sujeitos a interferências, sua confiabilidade a longo prazo é maior em comparação com os cabos, principalmente nos casos em que os dispositivos estão instalados em equipamentos com partes móveis, onde o desgaste dos cabos, devido ao movimento repetitivo, pode levar à falhas. Outra vantagem da eliminação de fios é nas aplicações em que instalar cabos é difícil, como por exemplo, aplicações caracterizadas por altas temperaturas, grandes altitudes e em máquinas rotativas (LUVISOTTO; PANG; DZUNG, 2017).

As aplicações industriais das redes sem fio são as mais variadas. A automação de processos, do inglês *Process Automation* (PA), se concentra no monitoramento e controle de processos químicos, biológicos ou outros em uma planta, normalmente distribuídos em uma grande área, envolvendo uma ampla variedade de sensores diferentes, que medem por exemplo, temperatura, pressão, fluxo e nível, e atuadores, como válvulas, aquecedores e bombas. A automação de fábrica, do inglês *Factory Automation* (FA) envolve por exemplo, linhas de montagem, controle de movimento de robôs, dispositivos móveis e tags de identificação. Os requisitos de desempenho variam com os casos de uso. Em geral, as aplicações de automação de fábrica tem requisitos mais rigorosos em termos de latência, confiabilidade e determinismo. Por exemplo, o controle de movimento de dispositivos robóticos pode exigir ciclos de comunicação menores que 1 ms enquanto que em PA, as aplicações podem tolerar latências maiores, como

é o caso de medições de temperatura e nível. Por outro lado, uma rede em FA, possui um número menor de nós (geralmente até 100), em comparação com as redes aplicadas a PA que cobrem todo o chão de fábrica com centenas até milhares de sensores distribuídos (LUVISOTTO; PANG; DZUNG, 2017), (5G-ACIA, 2019a).

Aplicações típicas de RSFI com gerenciamento centralizado envolvem a troca de dados cíclicos entre um controlador central e um conjunto de sensores e atuadores distribuídos. Estas redes são geralmente compostas por: gerenciador de rede, gateway, ponto de acesso e dispositivos de campo. O gerenciador é responsável por tarefas como, provisionamento de novos dispositivos, roteamento, escalonamento e otimização, além de adaptações a questões dinâmicas da rede (KÜNZEL; INDRUSIAK; PEREIRA, 2019). A Figura 1 apresenta um exemplo de uma RSFI.

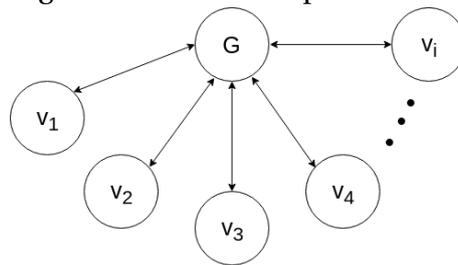


Fonte: do autor.

Uma RSFI pode ser modelada como um grafo  $G = (V, E)$  onde os nós  $v \in V$  representam os dispositivos de campo e o ponto de acesso,  $E$  é o conjunto de arestas entre esses dispositivos que representam os links (enlaces entre os dispositivos). A Figura 2 apresenta  $i$  dispositivos de campo conectados ao AP.

Em redes determinísticas, cada uma das entregas de ponta a ponta entre o ponto

Figura 2 – Topologia estrela com  $i$  dispositivos conectados ao AP.



Fonte: do autor.

de acesso e os dispositivos é definida como um fluxo  $F_i \in F$ , onde  $F = F_0, F_1, \dots, F_n$ . Cada fluxo  $F_i$  periodicamente gera um pacote com período  $P_i$ . Os períodos seguem uma cadeia de números múltiplos de acordo com a expressão  $ab^n$ , onde  $a$  e  $b$  são dois números constantes e  $n$  é um número natural qualquer, onde todos os períodos podem ser divididos entre si (NOBRE; SILVA; GUEDES, 2015). Exemplos de cadeias de períodos são 1, 2, 4, 8, 16 e 3, 6, 12, 24.

Para obter uma rede determinística, alguns protocolos utilizam o método de acesso ao meio *Time Division Multiple Access* (TDMA), onde o recurso de tempo é dividido em slots de tempo chamados de *timeslots*. Durante um *timeslot* ocorre a comunicação entre um par de dispositivos enquanto os demais permanecem em silêncio, evitando que colisões ocorram. Para organizar as comunicações, o gerenciador de rede precisa realizar o processo de escalonamento que consiste em reservar os recursos da rede para que as transmissões aconteçam. Os dispositivos de campo solicitam ao gerenciador recursos para que possam publicar seus dados. O gerenciador verifica se existem recursos disponíveis, e caso existam, aloca links nos *timeslots* para que os dispositivos possam enviar as mensagens. As taxas de publicação dos dados dependem da aplicação à qual o dispositivo está inserido, podendo estar na faixa de segundos à minutos em PA e de microssegundos à segundos em aplicações de FA (LUVISOTTO; PANG; DZUNG, 2017).

A utilização de gerenciamento centralizado traz vantagens como simplificação do *hardware* e *firmware* dos dispositivos de campo. Além disso, são mais adequadas para atender os requisitos de tempo real para aplicações industriais, uma vez que todas as tarefas de configurações são centralizadas no gerenciador evitando conflitos entre os nós distribuídos (MÜLLER, 2012).

Por outro lado, uma vez que o gerenciador realiza todas as tarefas de gerenciamento mencionadas, pode ocorrer que o tempo de execução dos algoritmos de ge-

renciamento influenciem significativamente na tomada de decisão por parte do gerenciador. A utilização de algoritmos de roteamento, escalonamento e otimização que realizem as tarefas de gerenciamento de forma rápida e eficiente é desejável para que a RSFI consiga atender as demandas das aplicações. O tempo em que o gerenciador leva para executar seus algoritmos de escalonamento deve ser avaliado e minimizado para que o gerenciador apresente respostas rápidas às alterações que ocorrem durante o funcionamento da rede, principalmente em aplicações do tipo FA, onde em determinados casos as taxas de publicação dos dispositivos são altas (maiores que 100 Hz) e os *timeslots* podem ter duração menor que 1 ms.

Em aplicações com alta taxa de publicação de dados, manter um escalonamento fixo pode ser uma boa opção para maximizar o determinismo e evitar sobrecarga. Por outro lado, pode ser necessário que o escalonamento seja alterado para atender a dinâmica da rede em eventos como entrada e saída de dispositivos, reconfigurações e otimização. Neste contexto são necessários estudos para minimizar a sobrecarga causada à rede pelos algoritmos de escalonamento (LUVISOTTO; PANG; DZUNG, 2019).

## 1.1 Motivação

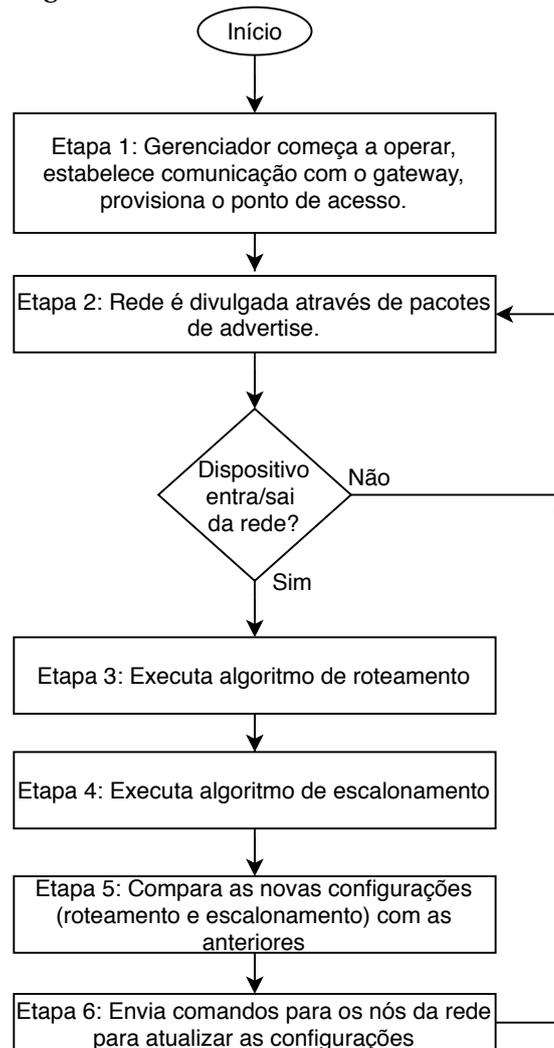
Ambientes onde as RSFI são instaladas apresentam condições adversas para dispositivos sem fio como por exemplo bloqueios físicos e interferências. A partir disto, a estabilidade da rede pode ser comprometida, uma vez que estes fatores podem reduzir a qualidade dos enlaces de comunicação. Neste caso, podem ser necessárias atualizações contínuas no roteamento e no escalonamento, visando superar estes problemas (R. TELES HERMETO; THEOLEYRE, 2017).

Em uma RSFI típica o algoritmo de escalonamento deve alocar gradualmente a largura de banda, dispositivo por dispositivo, uma vez que os dispositivos não entram todos ao mesmo tempo na rede. Uma alocação gradual pode se tornar não ótima e processos de otimização podem ser aplicados para melhorar o desempenho da rede. Este tipo de processo pode, por exemplo, fazer a realocação dos links de comunicação, para otimizar a ocupação da banda ou ainda gerar e testar novas configurações de escalonamento e roteamento. Em ambos os casos, os algoritmos de roteamento e escalonamento serão executados repetidas vezes (R. TELES HERMETO; THEOLEYRE, 2017). Além disso, em um gerenciador de rede real, restrições de hardware possivel-

mente ocorrerão, logo, é necessário que os algoritmos de escalonamento sejam de baixa complexidade.

A Figura 3 apresenta de forma simplificada a rotina de funcionamento de um gerenciador de rede em uma RSFI.

Figura 3 – Rotina do Gerenciador de Rede.



Fonte: do autor.

No momento em que o gerenciador começa a operar é estabelecida uma comunicação com o gateway e ocorre o provisionamento do ponto de acesso (Etapa 1). Assim que as configurações iniciais forem realizadas, o ponto de acesso começa a publicar pacotes de anúncio para que os dispositivos de campo entrem na rede (Etapa 2). Durante o processo de agregação dos dispositivos são executados os algoritmos de roteamento que constroem os grafos de broadcast (do gateway para todos os nós), *uplink* (dos nós para o gateway) e *downlink* (do gateway para os nós) (Etapa 3). Os grafos são

construídos levando-se em conta a topologia da rede. Para redes com topologia estrela, onde a redundância de caminhos não existe, o processo de roteamento não precisa ser executado. Em seguida, os algoritmos de escalonamento alocam os links nos *timeslots* (Etapa 4), levando em conta requisitos e regras da aplicação. Para evitar o envio de configurações redundantes, são comparadas as novas configurações geradas com as antigas (caso existam) (Etapa 5). Finalmente, as configurações são então convertidas em uma sequência de comandos e enviadas aos dispositivos (Etapa 6). Para realizar a reconfiguração, os novos links devem ser enviados aos dispositivos e somente então os antigos são removidos (POURIA *et al.*, 2014) (KÜNZEL; INDRUSIAK; PEREIRA, 2019). O gerenciador monitora a rede frequentemente e caso alguma alteração na topologia ocorra, i.e. entrada e saída de dispositivos, os algoritmos de roteamento e escalonamento são executados novamente.

Isso mostra a quantidade de tarefas que o gerenciador de rede deve atender durante a operação em uma RSFI. Conforme a rede aumenta, os processos tornam-se cada vez mais complexos e lentos, sendo assim a utilização de algoritmos que realizem o escalonamento de forma rápida pode tornar o gerenciador de rede mais eficiente e habilitar o uso deste tipo de rede para aplicações de FA. Em FA os dispositivos possuem taxas de atualização na faixa de microssegundos à milissegundos. Além disso, a ocorrência de dispositivos móveis é maior se comparado à PA.

Diversos trabalhos exploram estratégias de escalonamento para redes aplicadas a PA. Os objetivos principais são a diminuição da latência, diminuição do *jitter* e escalonabilidade. Em aplicações de FA, o escalonamento se torna ainda mais crítico, pois os tempos de ciclo estão na faixa de micro ou milissegundos. E desta forma, o tempo que o gerenciador leva para gerar o escalonamento pode representar um tempo alto em relação ao tempo de ciclo. Além disso em FA, a ocorrência de dispositivos móveis é maior e o escalonador deve convergir o mais rápido possível ao escalonamento viável.

Diante deste contexto, este trabalho visa aprofundar o entendimento de algoritmos de escalonamento no que se refere ao tempo de execução. Para tal, são comparados algoritmos de escalonamento de links e é apresentada uma proposta de algoritmo de escalonamento que visa reduzir o tempo de execução necessário para gerar tal escalonamento. São propostos estudos de caso que consideram a reconfiguração completa, onde todos os dispositivos são escalonados, e também a reconfiguração parcial,

quando um dispositivo agrega-se a uma rede já em funcionamento. Também é realizada uma comparação entre a utilização de múltiplos superframes (um para cada período de publicação) com a utilização de um superframe único.

## 1.2 Objetivos

O principal objetivo deste trabalho é realizar a análise de algoritmos de escalonamento de redes sem fios industriais visando aplicações de automação de fábrica.

Os objetivos específicos são:

- Verificar se a análise do tempo de execução dos algoritmos de escalonamento é relevante para o funcionamento de uma rede sem fio industrial;
- Implementar diferentes estratégias de escalonamento de links em redes TDMA;
- Propor uma estratégia de escalonamento que reduza o tempo necessário para gerar o escalonamento;
- Avaliar as estratégias de escalonamento em diferentes cenários;
- Desenvolver e implementar o processo de escalonamento em um gerenciador de rede;
- Avaliar as diferenças entre escalonamento com múltiplos superframes implementado no gerenciador proposto e com apenas um superframe utilizado por um gateway comercial.

A apresentação deste trabalho é organizada da seguinte forma: no Capítulo 2, são apresentados os conceitos básicos utilizados ao longo da dissertação. Um estudo sobre o estado da arte é apresentado no Capítulo 3. A metodologia utilizada e o algoritmo proposto neste trabalho são apresentados no Capítulo 4. Os estudos de caso são apresentados no Capítulo 5. Os resultados são apresentados e discutidos no Capítulo 6. Por fim, as conclusões deste trabalho e as sugestões de trabalhos futuros são apresentadas no Capítulo 7.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são discutidos conceitos relevantes para o entendimento desta dissertação. São apresentadas as áreas de aplicação de RSFI, casos de uso de RSFI em automação de fábrica, os protocolos de comunicação industrial para automação de fábrica e de processos e conceitos referentes ao escalonamento em RSFI.

### 2.1 Áreas de aplicação de RSFI

São diversas as áreas e casos de uso em que uma RSFI pode ser aplicada. Apesar de não se ter uma categorização explícita, é possível classificar algumas das aplicações nas seguintes áreas:

**Automação de Fábrica:** refere-se ao controle, otimização e monitoramento automatizado, incluindo linhas de produção, células de manufatura, controle de movimento de robôs e controle em malha fechada de sistemas de eventos discretos. Requer rigor temporal, em termos de latência e determinismo das comunicações. Algumas aplicações como processos de manufatura podem exigir latências menores que 10 *ms*. Engloba uma área geralmente menor do que 100 *m*<sup>2</sup>, sendo relativamente pequena se comparada à automação de processos (5G-ACIA, 2019a) (LUVISOTTO; PANG; DZUNG, 2019).

**Automação de Processos:** refere-se aos processos químicos, biológicos, petróleo e gás entre outras. Algumas variáveis de processo comuns neste tipo de aplicação são pressão, temperatura e nível. Os dados coletados pelos sensores são enviados a controladores centralizados que enviam os dados de controle para atuadores como, por exemplo, válvulas, bombas, aquecedores, fechando a malha de controle. Alguns dos requisitos são, a baixa latência, o determinismo e a confiabilidade. A área de abrangên-

cia é geralmente maior que em FA podendo variar de  $100m^2$  a vários  $km^2$  (5G-ACIA, 2019a) (LUVISOTTO; PANG; DZUNG, 2019).

**Interface homem-máquina (IHM) e vestíveis:** incluem diversos dispositivos para interação entre pessoas e o sistema de produção. As IHMs podem ser dispositivos como painéis, *displays*, *tablets* ou *smartphones* que permitam ao usuário interagir com a rede ou ainda painéis de controle de máquinas. Os dispositivos vestíveis são utilizados em aplicações tais como monitoramento de sinais fisiológicos, além da localização de um usuário na área de abrangência da rede. Além disso, os sistemas de realidade aumentada e virtual (AR/VR) que também fazem parte desta categoria apresentam potencial para desempenhar um papel cada vez mais importante no futuro (AGIWAL; ROY; SAXENA, 2016) (5G-ACIA, 2019a) (LUVISOTTO; PANG; DZUNG, 2019).

**Logística e Armazenamento:** refere-se à organização do armazenamento e do fluxo dos produtos e bens de consumo. Isso inclui armazéns automatizados, veículos guiados automaticamente, empilhadeiras entre outros dispositivos de manuseio de materiais. Alguns dos parâmetros mais relevantes neste tipo de aplicação são o posicionamento e o rastreamento dos materiais. Os requisitos mais importantes dos serviços de comunicação são a baixa latência, disponibilidade e determinismo. Engloba áreas internas e externas (5G-ACIA, 2019a) (LUVISOTTO; PANG; DZUNG, 2019) (LIANG *et al.*, 2019).

**Automação predial:** refere-se ao controle das operações que ocorrem dentro de imóveis como iluminação, segurança, gerenciamento de energia, ventilação e prevenção de incêndios (ZHU *et al.*, 2016).

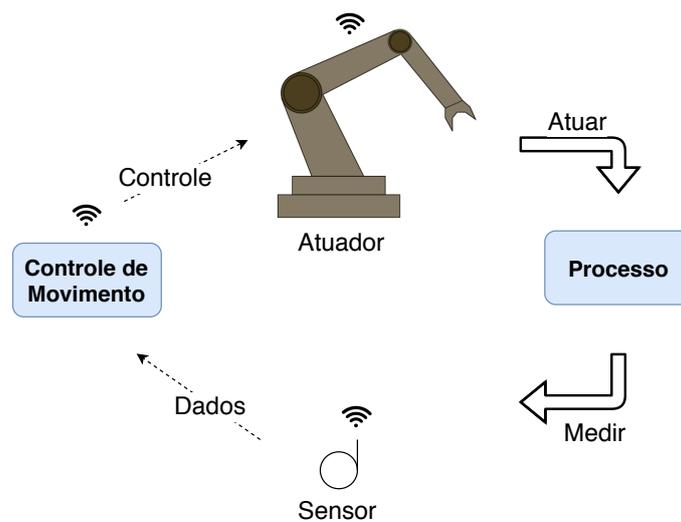
## 2.2 Casos de uso de RSFI em automação de Fábrica.

Os casos de uso no domínio industrial podem ser agrupados em diferentes categorias. Aqui são descritas algumas das categorias que envolvem processos característicos de FA.

**Controle de movimento:** é responsável pelo controle do movimento de partes móveis, muitas vezes rotativas, de uma máquina. O movimento repetitivo de um braço robótico, por exemplo, gradualmente gera desgaste nos fios e cabos instalados no dispositivo. Este tipo de equipamento, pode conter dezenas de cabos onde todos estão sujeitos a falha e uma fábrica pode conter centenas de dispositivos como este, logo os

custos de manutenção podem atingir valores consideravelmente altos (LIANG *et al.*, 2019). Esta aplicação é geralmente representada por um sistema em malha fechada, onde os sensores coletam dados do processo e enviam para o controlador de movimento. O controlador calcula a ação de controle e envia para um ou mais atuadores que executa a ação correspondente, como mostra a Figura 4.

Figura 4 – Exemplo de aplicação de uma RSFI no controle de movimento de um atuador robótico.



Fonte: do autor.

Nos casos onde alta precisão nos movimentos é necessária, o controle deve ser realizado de maneira estritamente cíclica e determinística. No trabalho de (LUVISOTTO; PANG; DZUNG, 2019) são apresentados alguns casos de uso encontrados na literatura onde os tempos de ciclo são menores que  $100 \mu s$ . Em (5G-ACIA, 2019b) são apresentados alguns casos de uso onde os tempos de ciclo podem ser inferiores a 2 ms como por exemplo, no caso de máquinas de embalagem e máquinas de impressão.

**Dispositivos móveis:** são equipamentos muito utilizados em movimentação de cargas. Podem ser equipamentos com algumas partes fixas como pontes rolantes, guinchos e esteiras ou dispositivos completamente móveis como Veículos Guiados Automaticamente, do inglês *Automated Guided Vehicles* (AGV). Um AGV é capaz de viajar por rotas pré-programadas para executar uma grande variedade de tarefas e é capaz de tomar decisões a partir do sensoriamento do ambiente. Os AGVs são robôs móveis empregados para manusear materiais dentro de uma determinada área. Para evitar que colisões entre os robôs aconteçam e para transmitir informações em tempo real, um sistema de controle deve monitorar os dispositivos. Este sistema é geralmente cen-

tralizado e os dispositivos comunicam-se com outros dispositivos como guindastes, pontes-rolantes, caminhões industriais, etc. Além disso, é necessário transmitir informações como, por exemplo, tarefas concluídas, materiais movimentados, localização e estado da bateria (5G-ACIA, 2019b). Em (GANGAKHEDKAR *et al.*, 2018) são apresentados alguns casos de uso como controle de movimento cooperativo de robôs, onde os tempos de ciclo são na casa de 1 *ms* e também aplicações de operação remota por vídeo, onde os tempos de ciclo são de 10 a 100 *ms*.

## 2.3 Protocolos de comunicação industrial

Diversos protocolos foram desenvolvidos para as aplicações de RSFI. Protocolos como o *WirelessHART*, ISA SP100.11a e WIA-PA visam atender os requisitos de PA. Estes protocolos possuem um comportamento lento durante a formação e manutenção da rede com tempos de configuração e envio de comandos variando na escala de segundos à minutos. Além disso, pela natureza dos processos aos quais estes protocolos atendem, os dispositivos são geralmente fixos e as taxas de atualização costumam ser maiores que 500 *ms* (POURIA *et al.*, 2014).

Algumas soluções buscam atender os requisitos de comunicação para as aplicações de FA, como é o caso do protocolo WISA/PNO WSAN (SCHEIBLE *et al.*, 2007), além do protocolo chinês WIA-FA que é a versão do WIA-PA para FA (LIANG *et al.*, 2019). Também existem algumas propostas recentes que podem se tornar soluções para as aplicações de FA como é o caso do *WirelessHP* e da próxima geração de comunicação móvel 5G (LUVISOTTO; PANG; DZUNG, 2019)(5G-ACIA, 2019b).

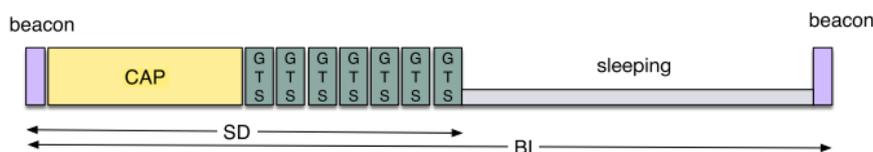
Abaixo, segue uma breve descrição de alguns protocolos que buscam atender aos requisitos de PA e FA.

### 2.3.1 IEEE802.15.4

Em 2003 a IEEE estabeleceu o padrão 802.15.4 visando aplicações industriais de controle e monitoramento, com baixo custo e baixo consumo de energia. A faixa de frequência utilizada pelo padrão é a *Industrial, Scientific and Medical* (ISM), que é de livre licença e opera na faixa de frequência de 2,4 a 2,483 GHz dividida em 16 canais. A taxa máxima de dados é de 250 kbit/s em um único canal. Dezesesseis canais são definidos, sendo que o último canal não pode ser utilizado globalmente.

A camada MAC utilizada no padrão combina dois métodos, um determinístico e um aleatório. O gerenciador de rede mantém seus vizinhos sincronizados, através do mecanismo de *beacon*. Os nós podem transmitir pacotes após a recepção do *beacon*, utilizando acesso aleatório no Período de Acesso à Contenção (CAP) ou em um *slot* agendado no período sem contenção (CFP). Na fase CAP, os nós compartilham um único *slot*, ou seja, qualquer nó pode transmitir e o método de acesso ao meio é o CSMA/CA. O período CFP é dividido em *timeslots*, e por esta razão é um período de garantia de acesso ao meio onde ocorre a entrega rápida e segura de pacotes de dados. Embora este período seja determinístico há apenas 7 *timeslots* garantidos, sendo esta uma severa restrição, dado que em uma aplicação industrial podem haver mensagens de um número elevado de sensores que deveriam apresentar algum tipo de garantia. A Figura 5 apresenta o superframe padrão da norma IEEE 802.15.4-2006 (R. TELES HERMETO; THEOLEYRE, 2017).

Figura 5 – Superframe no padrão IEEE 802.15.4-2006.



Fonte:(R. TELES HERMETO; THEOLEYRE, 2017).

### 2.3.2 WirelessHart, ISA SP100 e WIA-PA

O protocolo *WirelessHart* (WH) foi o primeiro padrão de comunicação sem fio aberto aprovado pela IEC para uso em controle e monitoramento de processos. O WH foi desenvolvido mantendo a compatibilidade com o padrão HART já bem estabelecido na indústria. O meio físico utilizado pelo protocolo WH é baseado na norma IEEE 802.15.4, desta forma a faixa de frequência utilizada é a ISM de 2,4 GHz com 15 canais. O método de acesso ao meio é TDMA, onde o tempo é dividido em *timeslots* de 10 ms. Além disso, para reduzir os efeitos de interferências externas o protocolo realiza o salto de canais juntamente com *blacklist*. O WH é adequado para aplicações de PA, uma vez que é determinístico, seguro e de baixo consumo de energia e o período mínimo de publicação suportado é de 250 ms (MÜLLER, 2012).

O ISA 100.11a também é um protocolo desenvolvido para aplicações de PA. Uma

das diferenças em relação ao WH é um número maior de especificações como por exemplo, políticas do gerenciador de segurança. Além de que, enquanto no WH o salto de canais possui uma única técnica, no ISA100 três técnicas diferentes podem ser aplicadas. Outra característica é que o *timeslot* pode ser ajustado de 10 ms até 14 ms (MÜLLER, 2012).

O protocolo WIA-PA (Wireless Networks for Industrial Process Automation) é o padrão chinês para RSFI. Em 2008 foi aprovado pela IEC, tornando-se o segundo padrão de comunicação sem fio industrial. O protocolo utiliza dois níveis de topologia de rede, sendo eles estrela e malha. Os dispositivos de campo formam uma rede em estrela onde o ponto central é um roteador. Os roteadores formam uma malha. As camadas física e MAC são baseadas na norma 802.15.4. A camada MAC possui períodos de contenção (CAP) e livre de contenção (CFP) e utiliza CSMA/CA como acesso ao meio. No período ativo, a porção de contenção é utilizada para agregar dispositivos à rede, gerenciamentos intra cluster e retransmissões. Já, a porção livre de contenção é utilizada para comunicação entre os dispositivos de campo e os *cluster heads* (VALADÃO *et al.*, 2018).

### 2.3.3 WISA/PNO WSAN

Em 2007 a ABB desenvolveu o protocolo Wireless Interface Sensor and Actuator (WISA), baseado na norma IEEE 802.15.1. Este protocolo foi desenvolvido com o objetivo específico de atender requisitos rigorosos das aplicações de FA. O WISA utiliza o método TDMA, com *timeslots* de 64  $\mu$ s para o donwlink e 128  $\mu$ s para o *uplink*. Cada estação base suporta até 120 sensores ou atuadores, e a duração total do superframe é fixa, 2048  $\mu$ s. Cada dispositivo pode transmitir em um dos 30 *timeslots* que compõem o superframe. A faixa de frequência é ISM de 2,4 GHz e também utiliza salto de canais (SCHEIBLE *et al.*, 2007) (ZAND *et al.*, 2012).

### 2.3.4 WirelessHP

O trabalho apresentado em (LUVISOTTO; PANG; DZUNG, 2017) mostra os requisitos de projeto para uma RSFI de alta performance para aplicações que requerem altas taxas de transmissão de dados, na faixa de Gbps e latências ultra baixas em torno de microssegundos.

Em (PANG; LUVISOTTO; DZUNG, 2017) é apresentada uma proposta seguindo es-

tes requisitos que é chamada de *High-Performance Wireless*, ou WirelessHP. O WirelessHP visa fornecer uma RSFI para aplicações críticas de controle industrial, como os casos de uso de FA. Este protocolo ainda é apenas uma proposta, não tendo sido padronizado até o presente momento. As metas básicas apontadas pelos autores são taxa de dados agregada de vários gigabytes por segundo, *timeslots* menores que  $1 \mu s$  e alto nível de confiabilidade. Para isso, o WirelessHP exige uma pilha de protocolos completamente personalizada. A fase atual da proposta está no desenvolvimento de uma camada física que busca reduzir o máximo possível o tempo de transmissão de pacotes. Uma das primeiras propostas para isso foram apresentadas em (LUVISOTTO *et al.*, 2017) e (LUVISOTTO; PANG; DZUNG, 2019) com base em uma versão modificada da camada física IEEE 802.11, utilizando multiplexação por divisão de frequências ortogonais (OFDM).

### 2.3.5 5G

A quinta geração de comunicação móvel, a 5G, é uma das principais candidatas para atender aplicações de FA. As características de altas taxas de transferência de dados e latências mais baixas do que as tecnologias LTE atuais, ampliaram os casos de uso onde este tipo de tecnologia pode ser utilizado. Algumas iniciativas, como a (5G-ACIA, 2019a) que reúne diversas empresas, pesquisadores e universidades alemãs, foram fundadas para explorar os aspectos necessários para que a 5G possa ser utilizada em aplicações industriais. Um dos objetivos do 5G é suportar comunicações ultra confiáveis de baixa latência (uRLLC), podendo transmitir pacotes com alta confiabilidade e latências de  $1 ms$ . Outra característica é a possibilidade de comunicações massivas entre máquinas (mMTC) que vai permitir o uso de milhares de dispositivos de campo em uma determinada área. Esta tecnologia promete abranger diversos casos de usos FA que requerem latências baixas (5G-ACIA, 2019a)(GANGAKHEDKAR *et al.*, 2018).

Os protocolos que buscam satisfazer requisitos específicos de aplicações de FA, como taxa de atualização menores que milissegundos, tentam reduzir tanto quanto possível a duração do *timeslot*. A Tabela 1 apresenta alguns dos protocolos desenvolvidos para aplicações em FA e PA.

Tabela 1 – Protocolos de comunicação industrial.

	Camada física	Acesso ao meio	Duração <i>timeslot</i>	Salto de canais	Num. de canais	Cenário alvo
WH	802.15.4	TDMA+ CSMA	10 ms	Sim	15	*PA
ISA100	802.15.4	TDMA + CSMA	10 a 14 ms	Sim	16	*PA
WIA-PA	802.15.4	TDMA + CSMA	10 ms	Sim	16	*PA
WIA-FA	802.11	TDMA (Adaptado)	100 $\mu$ s	Sim	3	**FA
Wisa/WSAN	802.15.1	TDMA + FDD	128 $\mu$ s UL 64 $\mu$ s DL	Sim	5	**FA
WirelessHP	802.11 (Adaptado)	TDMA + CSMA	1 $\mu$ s	Ainda não definido	Ainda não definido	**FA

\*Process Automation    \*\*Factory Automation

## 2.4 Escalonamento em Redes sem Fio Industriais

O escalonamento é o processo de atribuir cada tarefa de um conjunto de tarefas a um *timeslot*, sendo que o *timeslot* é a unidade básica de tempo em um sistema síncrono. O algoritmo de escalonamento tem a tarefa de definir a sequência das operações de forma que sejam executadas no menor tempo possível e de maneira correta. Em RSFI, onde a comunicação se dá através de links lógicos entre os dispositivos, a função do algoritmo de escalonamento é reservar *timeslots* para os links de forma a evitar conflitos.

O processo de escalonamento de links em RSFI é um problema NP-Difícil, uma vez que, para encontrar a melhor sequência de operações possível, o algoritmo precisa testar exaustivamente as configurações (ERGEN; VARAIYA, 2010), (SAIFULLAH *et al.*, 2010). O conjunto NP-Difícil representa problemas complexos de serem resolvidos, ou seja, podem existir infinitas soluções para resolver um problema tornando inviável a avaliação de cada uma delas em um tempo factível. Sendo assim, alguns estudos optam por tratar o problema de escalonamento através de uma abordagem baseada na utilização de heurísticas, em que é adotado um conjunto de procedimentos que visa a obtenção de uma resposta satisfatória para o problema, reduzindo consideravelmente o conjunto de soluções possíveis (NETTO, 1995).

Em 2012 foi lançada a emenda *e* do protocolo IEEE802.15.4e. Uma das características desta emenda é o método Time Slotted Channel Hopping (TSCH), que consiste

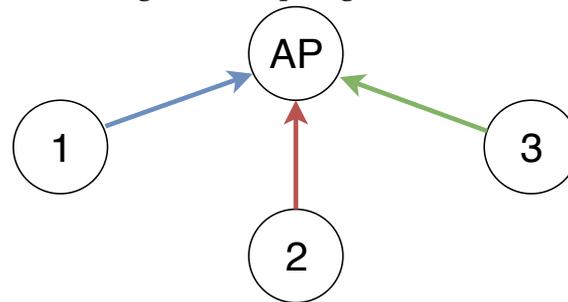
em dois conceitos: sincronização dos nós que resulta em determinismo e eficiência de energia, pois minimiza as colisões de pacotes, e o salto de canal que aumenta a confiabilidade (MIN *et al.*, 2016) (LUVISOTTO; PANG; DZUNG, 2019) (R. TELES HERMETO; THEOLEYRE, 2017).

Neste método o tempo é dividido em slots de tempo chamados de timeslots. Durante um *timeslot* ocorre a comunicação de um par de dispositivos enquanto os demais permanecem em silêncio evitando que colisões ocorram. A duração dos *timeslots* depende do protocolo utilizado e deve ter uma duração suficiente para que o transmissor envie o pacote e o receptor envie de volta uma confirmação de recebimento (ACK). Esse sistema pode ser representado por uma matriz  $M$ , na qual as linhas são referentes a frequência e as colunas o tempo. Cada célula é definida por intervalo de tempo e deslocamento de canal, indicando, respectivamente, seu instante de transmissão no superframe e a frequência que ela deve usar para a transmissão (R. TELES HERMETO; THEOLEYRE, 2017).

Os links de transmissão ou recepção são alocados nas células. Cada nó pode ter um ou mais links dependendo da quantidade de tráfego que precisa encaminhar. No TSCH são definidos dois tipos de células: dedicadas e compartilhadas. Células dedicadas são livres de contenção, isto é, são exclusivas para que um dispositivo realize a transmissão dos seus dados. Células compartilhadas podem ser utilizadas por dois ou mais dispositivos, utilizando um método de acesso ao meio aleatório, onde o pacote é transmitido e enquanto nenhum reconhecimento for recebido pelo transmissor informando que o receptor recebeu o pacote, presume-se que uma colisão ocorreu. Nesse caso, o transmissor realiza a retransmissão no próximo *timeslot* disponível (R. TELES HERMETO; THEOLEYRE, 2017).

Os *timeslots* da rede possuem uma estampa temporal chamada de Absolute Slot Number (ASN), que indica o número de intervalos de tempo desde o momento em que a rede foi iniciada (R. TELES HERMETO; THEOLEYRE, 2017). O ASN, idealmente, nunca se repete e por isso um mesmo *timeslot* terá diferentes valores de ASN nos ciclos subsequentes do superframe, isto é, o primeiro *timeslot* de um superframe vai ter um valor de ASN no primeiro ciclo, e outro valor nos demais ciclos. A Figura 6 apresenta um exemplo de topologia estrela onde os dispositivos de campo 1, 2 e 3 enviam seus dados para o ponto de acesso (AP).

Figura 6 – Topologia estrela.



Fonte: do autor.

A Figura 7 apresenta o escalonamento dos links dos dispositivos da Figura 6 no superframe. Neste exemplo o superframe é composto por 100 *timeslots* que se repetem ciclicamente. Percebe-se que a numeração dos *timeslots* referentes ao superframe é sempre a mesma, porém o ASN muda para um mesmo *timeslot* em um ciclo diferente do superframe.

Figura 7 – Matriz de escalonamento.

		Ciclo 0					Ciclo 1									
		Timeslot 0	1	2	3	4	...	99	0	1	2	3	4	...	99	...
Channel Offset	0	1->AP							1->AP							
	1		2->AP							2->AP						
	2						...							...		...
	...			3->AP							3->AP					
ASN		0	1	2	3	4	...	99	100	101	102	103	104	...	199	...

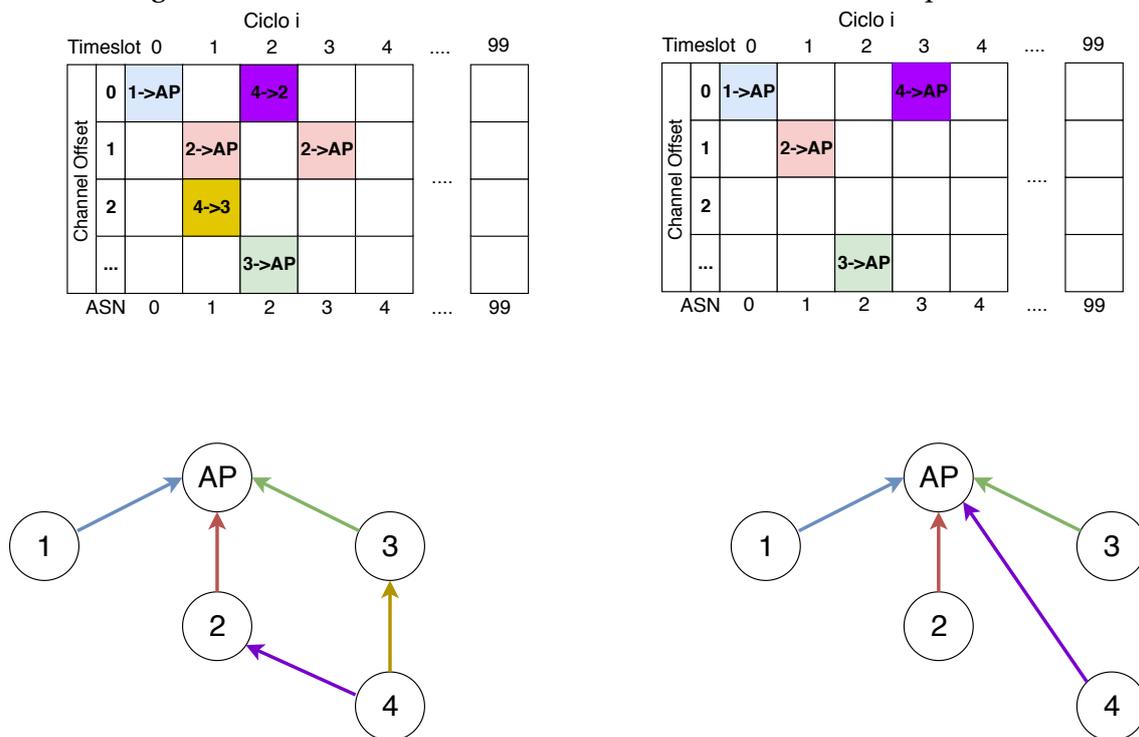
Fonte: adaptado de (R. TELES HERMETO; THEOLEYRE, 2017).

Em algumas aplicações de RSFI devem ser garantidos requisitos como alta confiabilidade, baixa latência e economia de energia. O método de acesso ao meio utilizado influencia diretamente nestes três requisitos. Para sistemas de tráfegos aperiódicos os métodos de acesso ao meio aleatórios como *Slotted Aloha* ou CSMA podem ser eficientes, uma vez que a perda de pacotes pode ser insignificante caso poucos dispositivos tentem disputar o meio, além de fornecerem baixa latência. No entanto, este tipo de técnica tende a falhar quando o sistema atinge um certo limite, onde muitos dispositivos tentam acessar o meio para transmitir seus dados e as colisões se tornam frequentes. Neste caso, métodos que organizem as transmissões em determinados instantes de tempo podem ser utilizados para evitar que ocorram colisões e consequentemente retransmissões, aumentando a confiabilidade e a economia de energia (R. TELES HER-

METO; THEOLEYRE, 2017).

Em redes do tipo *mesh*, dois links podem ser atribuídos ao mesmo *timeslot* desde que em canais não conflitantes o que é chamado de reuso de frequência. O mesmo não ocorre para redes *single-path* (topologia estrela) uma vez que o nó central participa de todas as comunicações que ocorrem na rede (DJUKIC; VALAEE, 2009). A Figura 8 apresenta estas duas situações. No exemplo da esquerda, o dispositivo 4 transmite para o 3 no mesmo *timeslot* que o dispositivo 2 transmite para o ponto de acesso. Para que isso seja possível os links possuem offsets de canal diferentes. O offset de canal é um valor utilizado no cálculo realizado pelo dispositivo para definir qual canal utilizar. Desta forma, os dispositivos utilizarão canais diferentes para realizarem as transmissões.

Figura 8 – Matriz escalonamento TSCH com e sem reuso frequência.



Fonte: do autor.

Através do salto de canais evita-se que as transmissões utilizem sempre o mesmo canal. As transmissões subsequentes de uma determinada transmissão seguem uma sequência pseudo-aleatória e portanto, cada link utiliza um canal diferente. Quando ocorre uma falha na transmissão, o próximo link disponível vai ser utilizado para retransmissão e utilizará um canal diferente diminuindo a probabilidade de ocorrer uma falha novamente. Desta forma o salto de canais melhora a confiabilidade em RSFI que sofrem interferências de redes externas (R. TELES HERMETO; THEOLEYRE, 2017). Um

nó utiliza a seguinte equação para calcular o índice do canal a ser utilizado:

$$I_{ch} = (Offset + ASN) \bmod(N_{Ch}) \quad (1)$$

$$Canal = Ch_{Ativos}[I_{ch}] \quad (2)$$

onde:

- *Offset* é referente ao deslocamento do canal;
- O *ASN* conta o número de *timeslots* desde o início da rede;
- $N_{Ch}$  é o número de canais ativos no mapa de canais;
- $Ch_{Ativos}$  vetor binário que ordena os canais que estão ativos no mapa de canais.

Depois que é calculado o  $I_{ch}$  por (1), o mesmo é utilizado como índice do vetor  $Ch_{Ativos}$  em (2).

Para aumentar a confiabilidade é importante utilizar toda a diversidade do espectro de frequência. Se o comprimento do superframe e o número de canais ativos no mapa de canais forem mutuamente primos, a mesma célula será mapeada para uma frequência diferente em ciclos diferentes do superframe, ou seja, o mesmo link utilizará frequências diferentes em cada ciclo do superframe.

Para evitar colisões, a rede deve ser sincronizada globalmente. A fonte de relógio principal da rede é o gateway. Cada nó possui sua fonte de relógio conforme seu número de saltos. Os nós que estão a um salto do gateway utilizam o próprio gateway como fonte de relógio. Os nós que estão a dois saltos do gateway utilizam como fonte de relógio os nós que estão a 1 salto do gateway e assim sucessivamente.

Uma vez que os nós são mantidos desligados a maior parte do tempo e se comunicam apenas nos *timeslots* agendados, pode se obter uma eficiência energética maior se comparado a protocolos com contenção. (MIN *et al.*, 2016) (LUVISOTTO; PANG; DZUNG, 2019).

Múltiplos superframes podem ser utilizados para organizar as comunicações. Podem ser definidos diferentes superframes para cada grupo de dispositivos baseando-se nos períodos de publicação de cada grupo. Por exemplo, em uma rede onde existem

três grupos de dispositivos, que publicam seus dados a cada 2 s, 4 s e 8 s, as transmissões de cada grupo podem ser organizadas dentro de três superframes diferentes, um para cada período. Além disso, os superframes podem ser adicionados, removidos e modificados durante o funcionamento da rede (MIN *et al.*, 2016).

Apesar dos superframes serem independentes uns dos outros, é necessário que todos estejam sincronizados, de tal forma que todos devem ter seus *timeslots* alinhados, mesmo que o início e o final de um ciclo específico do superframe não estejam. A Figura 9 apresenta um exemplo de dois superframes com tamanhos diferentes, o superframe 1 possui 8 *timeslots* enquanto que o superframe 2 possui 4 *timeslots*. Percebe-se que conforme os superframes se repetem, os *timeslots* ocorrem em ASNs diferentes, por exemplo, no primeiro ciclo do superframe 2 o *timeslot* 0 ocorre no mesmo ASN em que o *timeslot* 0 do superframe 1 ocorre. Já no segundo ciclo do superframe 2, o *timeslot* 0 ocorre no mesmo ASN em que o *timeslot* 4 do superframe 1 ocorre.

Figura 9 – Múltiplos superframes na rede.

Superframe 1 (8 timeslots)	TS0	TS1	TS2	TS3	TS4	TS5	TS6	TS7	.....
Superframe 2 (4 timeslots)	TS0	TS1	TS2	TS3	TS0	TS1	TS2	TS3	.....
ASN	0	1	2	3	4	5	6	7	

Fonte: adaptado de (MIN *et al.*, 2016).

Os superframes devem coexistir na rede sem gerar conflitos de maneira que um link utilizado em um superframe não deve conflitar com um link de outro superframe. Para isso, todos os superframes devem ser organizados dentro da matriz  $M$  de forma que não ocupem os mesmos *timeslots*. Os superframes possuem tamanhos múltiplos entre si, sendo que o tamanho de um superframe é dado pela quantidade de *timeslots*. A matriz  $M$  deve possuir o tamanho do maior superframe sendo que os superframes menores se repetem dentro da matriz. Por exemplo, se o maior período de publicação dos dispositivos da rede é de 16 *timeslots*, logo  $M$  tem 16 *timeslots* e um superframe de 8 *timeslots* se repete 2 vezes dentro de  $M$ .

A Figura 10 apresenta um exemplo onde os dispositivos  $v_1$ ,  $v_2$  e  $v_3$  da Figura 2 publicam seus dados periodicamente a cada 4, 8 e 16 *timeslots* respectivamente. A figura apresenta os superframes para cada um dos períodos mencionados no exemplo. Em cada um dos superframes é alocado um link de *uplink* que o dispositivo utiliza para

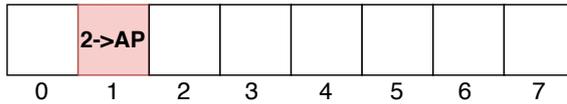
enviar seus dados ao ponto de acesso (AP).

Figura 10 – Superframes para períodos de 4, 8 e 16 timeslots.

S1 = 4 timeslots



S2 = 8 timeslots



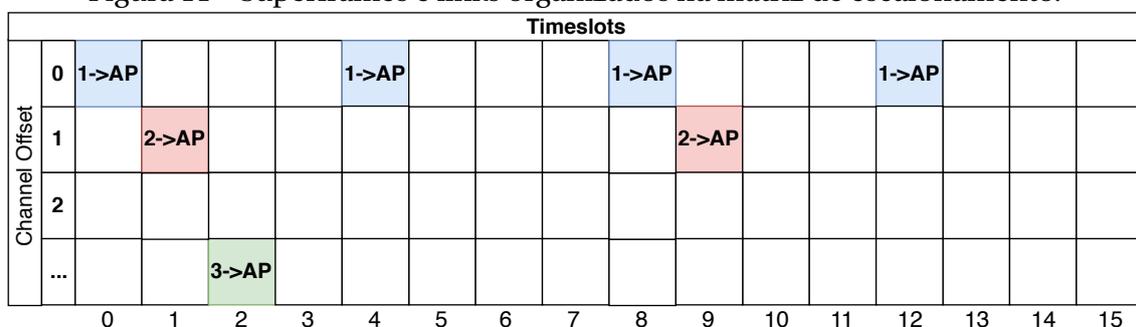
S3 = 16 timeslots



Fonte: do autor.

Apesar da possibilidade de organizar os links de comunicação em diferentes superframes, todos compartilham o mesmo recurso de tempo. Desta maneira os links devem ser organizados de forma que não ocorram colisões. É necessário avaliar também a matriz de escalonamento durante o escalonamento de links e não apenas os superframes. A Figura 11 apresenta os links dos 3 superframes organizados dentro da matriz de escalonamento. Percebe-se que os superframes compartilham a matriz de escalonamento  $M$  e que  $S_1$  e  $S_2$  se repetem mais de uma vez dentro de  $M$ . Uma vez que os superframes se repetem ciclicamente é suficiente encontrar um escalonamento para transmissões de pacotes gerados até o *timeslot*  $T$  (neste exemplo  $T = 16$ ).

Figura 11 – Superframes e links organizados na matriz de escalonamento.



Fonte: do autor.

### 3 ANÁLISE DO ESTADO DA ARTE

São diversos os casos de uso nos quais as RSFI podem ser aplicadas. Cada caso tem seus requisitos mais relevantes, tais como confiabilidade, latência e taxa de publicação de dados. A partir disto, para se obter um bom desempenho da RSFI é necessário adotar uma solução adequada para cada caso. Nesse contexto, as estratégias de roteamento e escalonamento adotadas têm um impacto direto no desempenho da RSFI. As propostas para o gerenciamento em RSFI são variadas, cada qual com seus objetivos e aplicações, e a comparação destas propostas pode auxiliar na escolha da mais adequada para cada caso.

Nesta capítulo será dada ênfase às técnicas de escalonamento, apesar de que o roteamento também é objeto de estudo pela comunidade científica. Um estudo aprofundado sobre o roteamento em RSFI é apresentado em (KÜNZEL; INDRUSIAK; PEREIRA, 2019) onde os autores aplicam técnicas de roteamento que fazem uso de aprendizado por reforço para ajustar as rotas de acordo com as condições operacionais atuais da rede. Os autores também apresentam algoritmos do estado da arte e comparam com a técnica proposta. Este trabalho pode ser considerado o estado da arte das técnicas de roteamento para RSFI e portanto pode ser o ponto de partida para o estudo do roteamento. A partir daqui serão discutidos apenas os trabalhos relacionados a escalonamento.

Em (DICKOW, 2014) e em (NOBRE; SILVA; GUEDES, 2015) são apresentados os principais algoritmos de roteamento e escalonamento para o protocolo WH. Os autores também fazem uma revisão de algoritmos de escalonamento que não são especificamente do WH. São apresentadas as principais características dos algoritmos de escalonamento com o objetivo de auxiliar na compreensão de uma implementação em RSFI. Os algoritmos são comparados pelos objetivos que possuem, pelas métricas

que utilizam, pela forma como foram validados (simulação ou em uma rede real), pelo método de redundância utilizado e finalmente se trabalham com múltiplos superframes ou não. Os objetivos de cada técnica são aquilo que os algoritmos pretendem melhorar quando utilizados em uma rede. Segundo os autores, todos os algoritmos tem um objetivo comum que é preencher a matriz de escalonamento, uma vez que este problema já é naturalmente complexo. Alguns algoritmos vão além e buscam atender outros requisitos melhorando alguma característica específica da rede como latência, confiabilidade e uso de recursos. Outra característica observada pelos autores é que alguns algoritmos de escalonamento possuem suporte para múltiplos superframes e outros não. A utilização de múltiplos superframes apresenta vantagens para as aplicações que possuem diferentes taxas de publicação. Uma característica dos algoritmos que suportam múltiplos superframe é que dispositivos com a mesma taxa de publicação são agrupados em um mesmo superframe. Outro fator observado é a redundância, que tem impacto direto na confiabilidade da rede. Uma forma de redundância utilizada pode ser considerando múltiplas rotas em redes do tipo malha ou ainda pelo uso de links de retransmissão em redes do tipo estrela.

Dentre todos os algoritmos apresentados em (DICKOW, 2014) e em (NOBRE; SILVA; GUEDES, 2015), dois dos mais citados na literatura são o de (HAN *et al.*, 2011) e o de (SAIFULLAH *et al.*, 2010). O trabalho de (SAIFULLAH *et al.*, 2010) implementa uma modificação no algoritmo Least Laxit First (LLF) e utiliza o deadline da mensagem como variável de decisão para pacotes que conflitem no mesmo *timeslot*. Este é um dos poucos trabalhos que apresenta o tempo de execução como uma das métricas de avaliação do algoritmo de escalonamento. Entre os dois, apenas o algoritmo de Han utiliza múltiplos superframes. O fato de suportar a utilização de múltiplos superframes e de garantir confiabilidade pelo uso de retransmissão destaca o algoritmo de (HAN *et al.*, 2011) dos demais. Este algoritmo é detalhado nas seções seguintes.

Algumas técnicas diferenciam os fluxos de dados e controle, sendo dados como *uplink* (comunicações dos dispositivos de campo para o gateway) e *downlink* (do gateway para os dispositivos de campo). Apenas o algoritmo (HAN *et al.*, 2011) diferencia os fluxos definindo qual região do superframe é ocupada por cada tipo de fluxo. O algoritmo determina que a primeira metade de cada superframe é dedicada para os fluxos de *uplink* e a segunda para os fluxos de *downlink*.

Os autores indicam que pesquisas futuras podem ser feitas em relação a novas métricas que poderiam ser utilizadas e também ao uso de *timeslots* compartilhados. Além disso, é chamada a atenção para a questão da escalabilidade. Quanto maior a rede mais ela é afetada pelo *overhead* dos algoritmos de roteamento e de escalonamento. Logo, os algoritmos devem ser executados o mais rápido possível para que o gerenciador consiga atender as mudanças na rede.

Em (R. TELES HERMETO; THEOLEYRE, 2017) são categorizadas inúmeras técnicas de escalonamento existentes de acordo com seus objetivos e suas características. Também são identificados alguns caminhos para trabalhos futuros que devem atrair a atenção dos pesquisadores nos próximos anos. Os autores distinguem duas categorias de soluções como centralizada e distribuídas. Cada técnica é classificada conforme as principais características dos cenários as quais são aplicadas, como por exemplo, tráfego dinâmico versus estático, links com perdas versus ideais, topologias móveis versus estáticas. Os autores listam sistematicamente as características suportadas por cada algoritmo de escalonamento por exemplo, dispositivos intermitentes, multirádios e modelos de interferência, fornecendo a base de pesquisa para a escolha do algoritmo de escalonamento para os engenheiros e pesquisadores desta área. Após a apresentação das técnicas existentes e suas respectivas aplicações, são indicadas algumas lacunas que podem ser exploradas em relação ao escalonamento. Uma delas é a utilização de estratégias híbridas centralizada e distribuída. Os autores sugerem que o algoritmo de escalonamento centralizado poderia distribuir a largura de banda para uma determinada área geográfica, enquanto a parte distribuída tenta alocar os intervalos de tempo determinados localmente. Esta abordagem traz consigo alguns problemas como interferências entre os próprios nós da rede e questões de hierarquia entre o escalonamento gerado de forma descentralizada com o escalonamento gerado pelo gerenciador de rede.

No trabalho de (MODEKURTHY; SAIFULLAH; MADRIA, 2019) é proposto um sistema de escalonamento distribuído em tempo real para redes WH. Para o escalonamento dos links de *uplink* o gerenciador de rede aloca dois *timeslots* dedicados para cada dispositivo na rota principal, um de transmissão e um de retransmissão. Um terceiro link é alocado em um caminho alternativo. A técnica é aplicável à atribuição de prioridade fixa ou dinâmica. Na prática, os fluxos podem ser priorizados com base

em prazos, períodos ou criticalidade. Foram considerados os algoritmos de prioridade fixa e de prioridade dinâmica *Deadline Monotonic* (DM) e *Earliest Deadline First* (EDF) respectivamente como políticas de escalonamento. Além de simulações, os autores também apresentam experimentos com redes reais. As métricas de avaliação utilizadas foram a taxa de escalonabilidade que é definida como a razão entre os fluxos que foram alocados e o total de fluxos e o tempo de execução que é o tempo médio gasto para que todos os nós obtenham um escalonamento excluindo o tempo necessário para gerar as rotas.

Para resolver o problema de escalonamento e restringir o atraso do tráfego de dados em RSFI (MIN *et al.*, 2016) propõe um algoritmo de escalonamento com suporte a múltiplos superframes. O trabalho apresenta dois procedimentos principais: Calcular o conjunto de *timeslots* disponíveis e alocar os recursos de comunicação de acordo com o requisito de tráfego de dados. Neste trabalho os autores apresentam os resultados comparando uma implementação de um único superframe com múltiplos superframes e concluem que existe uma redução no consumo de energia com o uso de múltiplos superframes o que conseqüentemente, aumenta o tempo de funcionamento do dispositivo de campo. Os autores mostram que organizar os links de comunicação em uma rede utilizando múltiplos superframes resulta em um ganho em termos de consumo atual de energia, na mesma rede usando apenas um superframe. Isso se dá pelo fato de que quando o tamanho do superframe toma como base a taxa de varredura mais rápida, para medições feitas em uma taxa mais lenta, os transceptores associados serão energizados tão frequentemente quanto a taxa de atualização da medição mais rápida, aumentando assim o consumo de energia. Além da energia, a memória também é um recurso limitado. Usando múltiplos superframes, são escritos no dispositivo apenas os links necessários para realizar as transmissões de acordo com o período de publicação de dados assim, a ocupação da memória é reduzida.

Alguns autores propõem técnicas descentralizadas para o escalonamento de links em RSFI. Em (BHATIA; HANSDAH, 2013) e em (MODEKURTHY; SAIFULLAH; MADRIA, 2019) são apresentadas técnicas de escalonamento descentralizadas. A essência destas estratégias é adotar o escalonamento local (no nível do dispositivo) permitindo que cada dispositivo agende suas transmissões usando uma política de escalonamento em tempo real, local e online. Este método evita a necessidade de criar e disseminar um

escalonamento global o que, segundo os autores reduz significativamente o uso de recursos. Os autores avaliam a escalonabilidade, o consumo de energia, a memória dos dispositivos e o tempo de execução e mostram que em comparação com estratégias centralizadas possui ganhos no tempo de execução dos algoritmos.

Algoritmos de escalonamento que visam alocar recursos para dispositivos móveis são apresentados em abordagens descentralizadas como em (MÜLLER *et al.*, 2013), e em abordagens centralizadas como em (DEZFOULI; RADI; CHIPARA, 2016), (DEZFOULI; RADI; CHIPARA, 2017) and (AL-NIDAWI; YAHYA; KEMP, 2016). Em (MÜLLER *et al.*, 2013) é proposta a utilização de coprocessadores, que são dispositivos com um nível de inteligência maior do que os dispositivos comuns, e que por isso, permitem o gerenciamento descentralizado. O coprocessador auxilia durante o evento de agregação de dispositivos intermitentes modificando o escalonamento das mensagens dinamicamente. Os estudos de caso apresentados, mostram que o gerenciamento local resulta em reduções de tempos de agregação, uma vez que o processo de escalonamento é transferido do gerenciador central para o coprocessador.

A abordagem apresentada em (DEZFOULI; RADI; CHIPARA, 2016) é capaz de atender dispositivos móveis de maneira centralizada. A proposta apresenta uma rede multi saltos com dispositivos móveis e dispositivos fixos. É apresentado um algoritmo de escalonamento chamado Mobility Aware Scheduling Algorithm (MASA). O trabalho é apresentado sob o contexto de WH. A principal vantagem deste método em relação aos demais é que segundo os autores, o MASA consegue utilizar de maneira mais eficiente a banda disponível quando a mobilidade é introduzida. Algumas das métricas utilizadas para avaliar o algoritmo foram a escalonabilidade, latência média, tempo de vida dos dispositivos, tempo de execução e o tempo no processo de agregação.

Uma das lacunas encontradas nos trabalhos revisados é que os algoritmos de escalonamento não avaliam casos aplicados a automação de fábrica onde, devido a alta taxa de publicação de dados dos dispositivos, o tempo de execução dos algoritmos pode ser uma métrica relevante de ser avaliada. As principais métricas que os trabalhos avaliam são a taxa de sucesso como em (HAN *et al.*, 2011) e em (NOBRE; SILVA; GUEDES, 2015). O tempo de execução nem sempre é avaliado, e quando é, como no caso de (SAIFULLAH *et al.*, 2010), é no âmbito de uma aplicação de automação de processos. Portanto, este trabalho se diferencia dos demais por apresentar uma com-

paração entre estratégias de escalonamento de links para RSFI e por propor uma estratégia de escalonamento centralizado que visa reduzir o tempo do processo de escalonamento. Além disso, são comparadas as estratégias de utilização de um único superframe com tamanho fixo com a estratégia de uso de múltiplos superframes. A Tabela 2 apresenta uma breve comparação entre os principais trabalhos analisados.

Tabela 2 – Principais trabalhos analisados.

	Compara Algoritmos	Propõe Algoritmo/Técnica	Centralizado	Múltiplos Superframes	Tempo de execução
(SAIFULLAH <i>et al.</i> , 2010)		✓	✓		✓
(HAN <i>et al.</i> , 2011)		✓	✓	✓	
[Bhatia <i>et al.</i> , 2013]		✓			✓
[Müller, 2013]		✓			
(DICKOW, 2014)	✓				
(NOBRE; SILVA; GUEDES, 2015)	✓				
(R. TELES HERMETO; THEOLEYRE, 2017)	✓				
(MIN <i>et al.</i> , 2016)		✓	✓	✓	
[Modekurthy <i>et al.</i> , 2019]		✓			✓
[Cainelli, 2020]	✓	✓	✓	✓	✓

## 4 MATERIAIS E MÉTODOS

O presente capítulo apresenta os materiais e métodos utilizados no desenvolvimento deste trabalho.

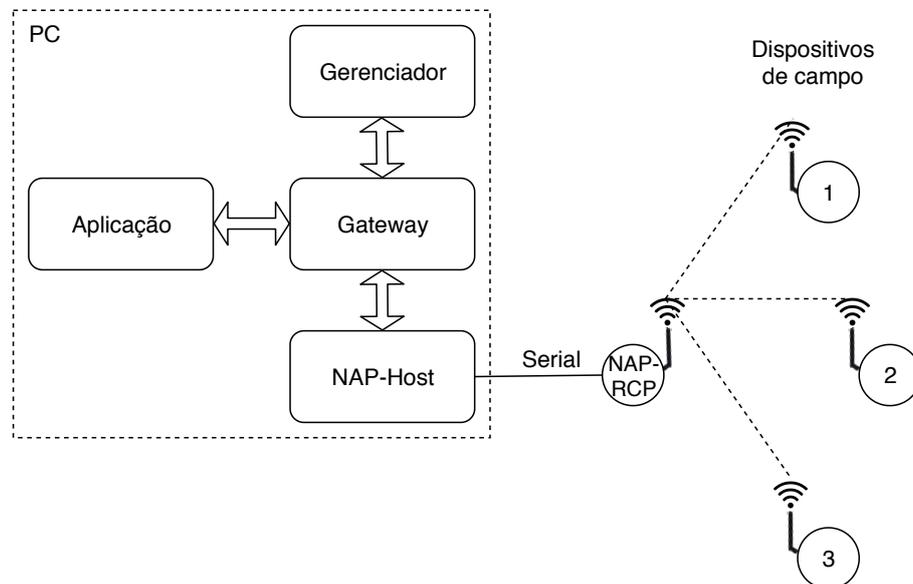
### 4.1 Materiais

Este trabalho fez uso de um gerenciador de rede desenvolvido em (RECH, 2012) e propiciou, através da continuidade, o desenvolvimento do processo de escalonamento no gerenciador. O gerenciador segue o protocolo WH e é composto basicamente por três processos executados em um sistema Linux: o gerenciador de rede, o gateway, que é o processo responsável por encaminhar os pacotes e interpretar os comandos do protocolo e o host do ponto de acesso (NAP-host), que se comunica com o ponto de acesso físico (NAP-RCP) através de uma porta USB.

Os códigos foram implementados nas linguagens de programação C e C++, estruturados de forma modular, orientado a objetos e são executados no sistema operacional Linux. A comunicação entre os processos se dá através da biblioteca *libchan*. A *libchan* é uma abstração da comunicação via socket UNIX e é visto pelas entidades como um "canal de comunicação" (RECH, 2012).

A Figura 12 apresenta a arquitetura do sistema onde um quarto elemento é apresentado que é a aplicação. A aplicação pode ser qualquer software capaz de se comunicar com o gateway WH. Neste caso a aplicação foi desenvolvida para envio de comandos aos dispositivos.

Figura 12 – Diagrama da arquitetura de rede empregada, de acordo com o protocolo WH.



Fonte: adaptado de (MÜLLER, 2012).

A Figura 13 apresenta quatro terminais onde estão sendo executados o gerenciador, o gateway, o NAP-Host e a aplicação.

Figura 13 – Terminal do PC com o gerenciador, o gateway, o NAP-Host e a aplicação.

Gerenciador	Gateway	NAP-Host	Aplicação
<pre> TRAFFIC to NETDEVICE 0x1234 (send [02] [00] [3D] [02] [77] [00] [00] [00] [00] [02] [12] [34] [56] [A7] [00] [00] [00] [05] [00] [00] [02] [08] [00] [00] [03] [05] [00] [00] [C7] [08] [00] [00] [06] [05] [00] [03] [C7] [08] [01] [00] [1E] [05] [02]  TRAFFIC from GATEWAY (net_proc) CMD NETDEVICE NICKNAME = 0x1234 [02] [00] [0F] [03] [77] [00] [00] [00] [00] [02] [12] [34] [03] [C7] [00] [00] [05] [00] [00] [02] [03] [C7] [0B] [00] [00] [00] [03] [05] [03] [00] [3E] [03] [C7] [0B] [00] [05] [00] [00] [05] [03] [00] [3D] [00] [01] [00] [1E] [05] [FF] [FF] [3C] </pre>	<pre> TRAFFIC to NETWORK MANAGER (NETMGR_Tr NICKNAME = 0x1234 [02] [00] [0F] [03] [77] [00] [00] [0 [00] [00] [02] [12] [34] [03] [C5] [0 [00] [64] [01] [0F] [03] [C5] [06] [0 [64] [01] [0E] [03] [C5] [06] [00] [0 [01] [0D] [03] [C5] [06] [00] [03] [0 [0C] [03] [C5] [06] [00] [04] [00] [C [03] [C5] [06] [00] [05] [01] [90] [0 [C5] [06] [00] [06] [03] [20] [01] [0 [06] [00] [07] [06] [40] [01] [08]  ===== function: gateway_sniffer --&gt; cmd parser cmd: 965 len: 6 blen: 72 rc 0 cmd: 965 len: 6 blen: 63 rc 0 cmd: 965 len: 6 blen: 54 rc 0 cmd: 965 len: 6 blen: 45 rc 0 cmd: 965 len: 6 blen: 36 rc 0 cmd: 965 len: 6 blen: 27 rc 0 cmd: 965 len: 6 blen: 18 rc 0 cmd: 965 len: 6 blen: 9 rc 0 ===== </pre>	<pre> TRAFFIC to NAP-RCP (nap_proc_gtw) [7E] [7E] [00] [00] [4F] [00] [00] [ [00] [00] [03] [00] [00] [03] [00] [ [11] [B9] [03] [00] [12] [34] [F9] [ [60] [6D] [C5] [4B] [B0] [1B] [2E] [ [47] [CC] [28] [57] [4C] [2D] [80] [ [2D] [CE] [A5] [E7] [E1] [7F] [A5] [ [DF] [C6] [99] [A4] [99] [26] [8C] [ [A0] [FA] [AF] [A2] [2F] [A5] [35] [ [FA] [24] [80] [7E]  TRAFFIC from NAP-RCP (nap_proc_rcp) [7E] [7E] [02] [03] [00] [1C] [59] [  TRAFFIC from NAP-RCP (nap_proc_rcp) [02] [03] [00] [1C] [59]  TRAFFIC to GATEWAY (nap_proc_rcp) - </pre>	<pre> -- Wireless Hart Command App -- ----- NICKNAME [HEX]: </pre>

Fonte: do autor.

#### 4.1.1 Gerenciador de rede

No gerenciador de rede estão as funções referentes à criação da rede e ao provisãoamento dos dispositivos. Algumas das classes que compõem o código do gerenciador são:

**Classe Netdevice:** classe referente aos dispositivos da rede. Cada dispositivo da rede é representado por um objeto desta classe. Contém atributos como nickname, tipo (gateway, NAP, gerenciador ou dispositivo de campo), período de publicação (para os dispositivos de campo), entre outros.

**Classe Superframe:** esta classe é referente aos superframes da rede. Contém atributos como id do superframe, número de timeslots, estado (ativo ou inativo) e uma lista encadeada de links que estão alocados neste superframe.

**Classe Link:** possui atributos como id do link, id do superframe no qual este link esta alocado, o número do slot em que o link esta alocado, o offset de canal, o tipo de link (Normal, Discovery, Broadcast ou Join) e a opção do link (transmissão, recepção ou compartilhado).

**Classe Session:** possui atributos como tipo, netdevice e contador. Esta classe é responsável por armazenar as informações referentes as sessões que o gerenciador cria entre os dispositivos para que as comunicações possam ocorrer.

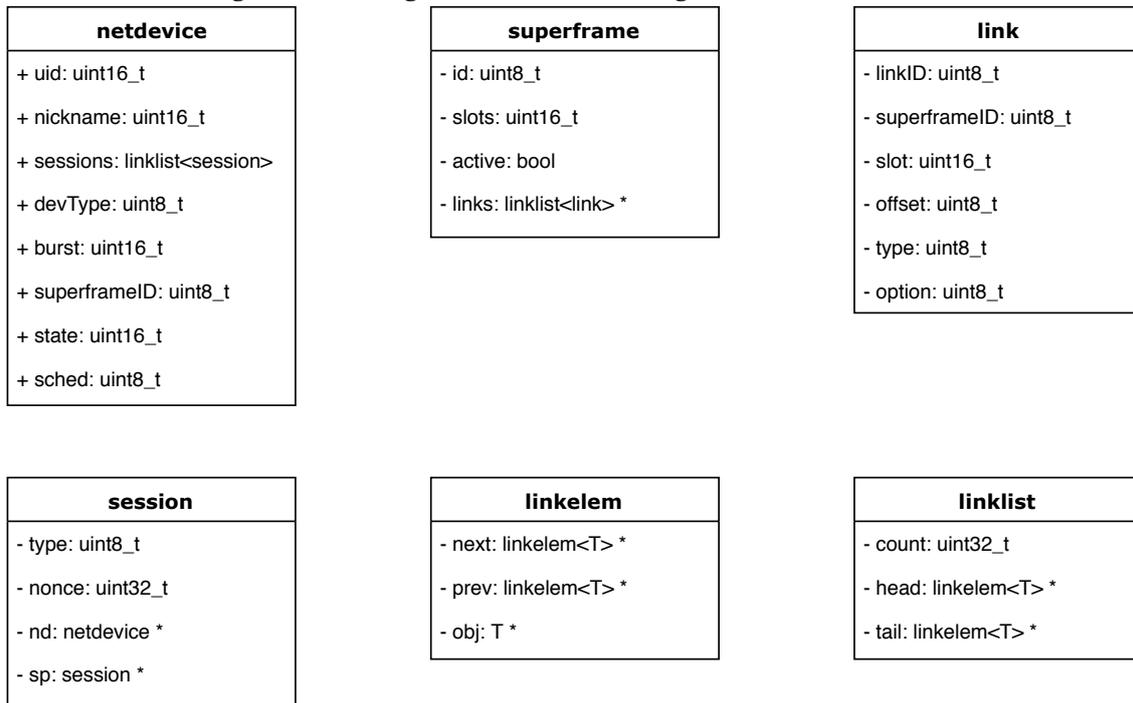
**Classes linkelem e linklist:** estas classes são utilizadas para a organização das listas de objetos criados durante a execução do código. A classe linklist implementa uma lista encadeada de elementos. A classe linkelem é referente a uma unidade da lista encadeada de elementos e contém um objeto.

A Figura 14 apresenta um diagrama com as classes mais importantes do gerenciador.

Diversas alterações foram necessárias para o correto funcionamento do gerenciador de rede, visto que a versão em que o gerenciador se encontrava, no início deste trabalho, não realizava o provisionamento de novos dispositivos na rede. Para isso foi necessário resolver diversas inconsistências presentes no código fonte. Assim que o gerenciador começou a operar, percebeu-se que o processo de escalonamento foi feito de forma provisória e não considerava a quantidade de links de dados e o tamanho dos superframes. Portanto, as principais contribuições foram nas funções de criação e manutenção da rede e também nas funções responsáveis pelo processo de agregação e escalonamento de novos dispositivos. Alguns dos principais arquivos utilizados no gerenciador que contém estas funções são:

**cfg\_netmgr.cc:** Neste arquivo, a função *configure\_netmgr()* é responsável por realizar as configurações iniciais da rede. São instanciados os objetos da classe netdevice

Figura 14 – Diagrama de classes do gerenciador de rede.



Fonte: do autor.

para o gerenciador, gateway e NAP. Além disso, são criados os links de Join que serão enviados ao NAP para que novos dispositivos possam entrar na rede. Na sequência são criados 10 superframes: um para os links de join com tamanho de 100 timeslots, um para links de gerenciamento com tamanho de 100 timeslots, e 8 para os links de dados com tamanhos variando de  $2^n$  sendo  $0 < n < 6$ . Também são construídas as listas encadeadas da classe *linklist* que armazenam os objetos referentes aos dispositivos da rede, aos links e aos superframes que posteriormente serão utilizados no escalonamento. As listas construídas são mantidas em memória como instâncias da classe *linklist* e são definidas no arquivo *netobjs.cc*.

**formation.cc:** Neste arquivo constam funções de envio de mensagens para o gateway (*send\_to\_gateway()*) e para o netdevice (*send\_to\_netdevice()*). Além disso, existem funções para realizar o provisionamento do NAP (*nap\_formation()*) e dos dispositivos de campo (*fd\_formation()*). As principais contribuições deste trabalho para o desenvolvimento do gerenciador de rede ocorreram neste arquivo. Foi necessário alterar a sequência de envio de comandos para realização do provisionamento dos dispositivos dentro da função *fd\_formation()*. Esta função é formada principalmente por uma estrutura de *switch/case* onde em cada *case* são enviados comandos para o NAP

e para o dispositivo de campo. Os trabalhos anteriores não consideraram a realização do escalonamento de links de forma organizada.

**cmd\_processor.cc:** Neste arquivo, a função *cmd\_processor()* é responsável por direcionar para diferentes funções a mensagem recebida. Caso a origem da mensagem seja o Gateway a função chamada é a *cmd\_gateway()*, caso seja um dispositivo de campo, a função chamada é *cmd\_netdevice()* e caso a mensagem venha de uma aplicação externa à rede é chamada a função *cmd\_application()*.

#### 4.1.2 Gateway

O gateway é responsável por conectar os elementos da rede como o gerenciador, os dispositivos de campo e as aplicações da planta. Algumas das principais funções do código do gateway são as funções de leitura como *nap\_read()*, responsável por receber o tráfego que vem do NAP, a função *app\_read()* que recebe o tráfego da aplicação e *proc\_netmgr()* que faz a leitura do tráfego oriundo do gerenciador. O Gateway atua no direcionamento do tráfego, assim sendo, as mensagens que tem como destino os dispositivos de campo são enviadas pelo gateway ao ponto de acesso (NAP-RCP) através do host do ponto de acesso (NAP-host). Ao contrário, ocorre o mesmo, as mensagens que vem da rede para o gateway passam pelo NAP-RCP e posteriormente pelo NAP-host.

#### 4.1.3 NAP-Host e NAP-RCP

O ponto de acesso é formado pelo NAP-Host e pelo NAP-RCP. O NAP-Host é o software responsável por realizar a interface entre o ponto de acesso físico (NAP-RCP) e o gateway. As principais funções utilizadas são a *nap\_proc\_gtw()* que trata o tráfego oriundo do gateway e a função *nap\_proc\_rcp()* responsável por encaminhar o tráfego oriundo do ponto de acesso físico. O NAP-RCP é o ponto de acesso físico e consiste em um rádio MC1322X executando um *firmware* WH modificado.

Durante a inicialização, o NAP-host tenta se conectar ao gateway na rede. Após encontrar o gateway, o NAP-host conecta-se ao ponto de acesso físico através de uma conexão serial. Uma vez conectado ao gateway e ao NAP-RCP, o tráfego entre estas duas entidades pode iniciar. O NAP-Host monitora cada canal de conexão com os dois elementos (Gateway e NAP-RCP) chamando a respectiva função de processamento para cada tipo de mensagem recebida.

#### 4.1.4 Dispositivos de campo

Os dispositivos de campo são os rádios utilizados para conectar os sensores ou os atuadores à rede WH. O hardware utilizado foi desenvolvido em (MULLER *et al.*, 2010). Os dispositivos possuem a pilha do protocolo WH e são compostos por um microcontrolador ARM 32-bit com um transceptor de rádio IEEE 802.15.4 integrado além de periféricos como RTC, gerenciamento de energia e portas para comunicação cabeada. O *firmware* foi desenvolvido em linguagem C para sistemas embarcados restritos.

O hardware e o *firmware* utilizados foram inicialmente desenvolvidos para dispositivos de campo. Assim, para utilizar o dispositivo como ponto de acesso foram realizadas alterações no *firmware* em (HAHN, 2011), onde foram implementadas funções de recepção e transmissão das mensagens através da porta serial.

A Figura 15 apresenta o dispositivo de campo utilizado.

Figura 15 – Dispositivo de campo utilizado nos testes práticos.



Fonte: (MULLER *et al.*, 2010).

#### 4.1.5 Gateway Comercial

O gateway comercial Emerson Wireless 1420A Gateway (EMERSON, 2013), apresentado na Figura 16, foi utilizado para testes de comparação com o gerenciador desenvolvido neste trabalho. Este dispositivo é uma plataforma que integra o gerenciador de rede, o gateway e o ponto de acesso WH.

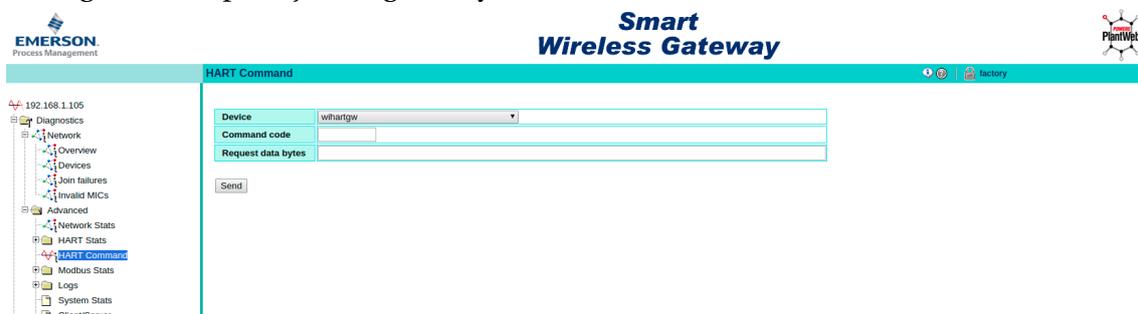
Figura 16 – Gateway comercial Emerson 1420A (EMERSON, 2013).



Fonte: (KRÖTZ, 2019).

O gateway possui uma interface onde é possível extrair informações e enviar comandos para os dispositivos. A interface é apresentada na Figura 17 e pode ser acessada por um computador através do IP do gateway.

Figura 17 – Aplicação do gateway comercial Emerson 1420A (EMERSON, 2013).



Fonte: do autor.

#### 4.1.6 Sniffer

Para realizar a captura dos pacotes durante os testes práticos, foi utilizado um sniffer compatível com o protocolo WH. O sniffer WiAnalis apresentado na Figura 18 é um dispositivo receptor de RF que captura os pacotes que trafegam na rede WH dentro de seu raio de alcance. O dispositivo é capaz de decifrar as mensagens e armazenar em seus registros informações dos pacotes como: ASN em que o pacote foi gerado, ASN em que ocorreu a transmissão, tempo/hora da transmissão, tipo de pacote (comandos, anúncio), origem e destinatário do pacote e informações contidas nos comandos.

Figura 18 – Sniffer WiAnalys utilizado para captura das mensagens durante os testes práticos.



Fonte: (KRÖTZ, 2019).

## 4.2 Métodos

Abaixo são apresentados os métodos utilizados no desenvolvimento deste trabalho. Inicialmente é apresentado o processo de escalonamento considerado. Em seguida é descrito o escalonamento com múltiplos superframes e o processo de agregação de um dispositivo de campo na rede. São apresentados os algoritmos de escalonamento implementados e por fim é descrita a proposta de algoritmo de escalonamento.

### 4.2.1 Processo de Escalonamento

Para o desenvolvimento do processo de escalonamento foi utilizado como base a norma WH por se tratar de um dos protocolos de comunicação industrial mais consolidados.

Apesar deste protocolo ter sido desenvolvido visando automação de processos, os conceitos utilizados no desenvolvimento do escalonamento podem ser aplicados também em automação de fábrica. Além disso, o grupo de pesquisa do Laboratório de Sistemas de Controle, Automação e Robótica da UFRGS (LASCAR), onde este trabalho foi desenvolvido, tem longa experiência neste protocolo possuindo todas as ferramentas necessárias para o desenvolvimento do trabalho.

Inicialmente foi utilizado o código do gerenciador de rede para realizar a implementação e teste dos algoritmos de escalonamento de forma *offline*. Nesta etapa, foram analisados o tempo de execução e a taxa de sucesso dos algoritmos de escalonamento. Ao final, os algoritmos foram integrados ao gerenciador e executados durante o funcionamento da rede para verificar o funcionamento do escalonamento com múltiplos

tiplos superframes e comparar com o processo de escalonamento gerado pelo gateway comercial.

A topologia considerada neste trabalho foi estrela, sendo assim os dispositivos conectam-se apenas com o NAP. Podem haver diferentes períodos de publicação dos dispositivos na rede sendo que para cada período é utilizado um superframe com o tamanho referente ao período.

A seguir são descritos o processo de escalonamento considerando a utilização de múltiplos superframes e também o processo de agregação de um dispositivo à rede.

#### **4.2.2 Escalonamento com múltiplos superframes**

A utilização de um único superframe possui como principal vantagem a simplicidade de implementação, porém isso implica na baixa eficiência de utilização dos recursos da rede. Com um único superframe, o tamanho deve ser definido pelo menor ou pelo maior período de publicação dos dispositivos da rede. Apesar de este método eliminar a necessidade de gerenciar múltiplos superframes, pois reduz a complexidade da tarefa de escalonamento, ele possui algumas desvantagens. Uma delas, no caso do comprimento ser o menor período é que os transceptores dos dispositivos associados aos períodos mais rápidos serão energizados com a mesma frequência que os de período mais lento, ocasionando um aumento no consumo de energia. Já, um superframe que tenha comprimento com base no maior período necessitará de mais *timeslots* reservados para comunicações que são realizadas em períodos mais curtos e conseqüentemente mais links serão necessários, aumentando a memória requerida para armazenar os links em um dispositivo (FOUNDATION, 2008). Por exemplo, se o superframe for ajustado com tamanho de 1000 *timeslots* e um dispositivo publica sua variável a cada 200 *timeslots*, este dispositivo vai precisar de 5 *timeslots* dedicados a ele dentro do superframe. Se o contrário ocorrer, e o superframe for ajustado com tamanho 200 *timeslots*, o dispositivo que publica a cada 1000 vai precisar de apenas um link, porém em 5 ciclos do superframe o dispositivo não vai publicar nada, consumindo recursos que poderiam ser utilizados por outros dispositivos. Posto isto, a utilização de múltiplos superframes (um para cada período) pode ser uma alternativa para evitar estes problemas.

Para validar os algoritmos de escalonamento em uma rede real foram implemen-

tados os algoritmos no gerenciador de rede desenvolvido. Como dito anteriormente o gerenciador de rede segue a norma WH. A norma WH não especifica qual algoritmo de escalonamento deve ser utilizado, mas ela descreve algumas questões que devem ser observadas. Uma delas é relacionada aos superframes. A norma diz que uma boa prática é organizar as comunicações em diferentes superframes como por exemplo, superframe de dados e de gerenciamento. Inicialmente o gerenciador utilizava apenas um superframe para escalonar todas as transmissões. A fim de organizar a distribuição dos recursos foi determinado que os links de anúncio (Join) seriam alocados em um superframe específico diferente do superframe de gerenciamento e de dados. No superframe de dados estarão escalonados os links para transmissão de informações relacionadas aos dados do processo, já no superframe de gerenciamento estão os links relacionados ao envio de comandos de gerenciamento como por exemplo comandos relacionados a escrita de links ou relacionados ao estado dos dispositivos.

Para evitar qualquer conflito entre os *timeslots* reservados para envio das variáveis de processo e para comunicações de gerenciamento, o superframe de gerenciamento deve ter comprimento múltiplo do menor superframe de dados. Além disso devem ser observados que os *timeslots* não podem conflitar com os *timeslots* utilizados no superframe de dados (FOUNDATION, 2008).

Segundo a especificação da norma WH (FOUNDATION, 2008), apenas uma comunicação de gerenciamento é permitida por vez. Assim sendo, uma vez que apenas uma transmissão irá ocorrer por vez no superframe de gerenciamento, um dispositivo com vários pares pode ser agendado para receber dados no mesmo slot, pois a transmissão irá apenas para um dispositivo (FOUNDATION, 2008).

**Superframe de Gerenciamento:** o superframe de gerenciamento é destinado à troca de informações de gerenciamento, como por exemplo, comandos para configuração de dispositivos (FOUNDATION, 2008). Este é um superframe comum a todos os dispositivos da rede, porém os mesmos recebem apenas as partes do superframe diretamente associadas a eles.

A frequência com que as transmissões são agendadas nos superframes de gerenciamento determina o tempo de resposta das solicitações. Esta frequência é ajustada pelo tamanho do superframe. Por exemplo, considerando que cada *timeslot* tenha duração de 10 ms e que o superframe tenha o comprimento de 100 *timeslots*, a taxa máxima

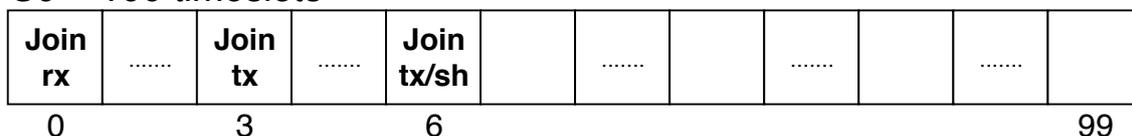
que solicitações e respostas são transferidas é 1 vez a cada segundo (FOUNDATION, 2008).

Para organizar as comunicações de gerenciamento, optou-se por utilizar dois superframes: um especificamente para o processo de agregação (join) e outro para o envio de comandos.

A Figura 19 apresenta o superframe de anúncio. Neste superframe foram alocados inicialmente três links de join utilizados pelo NAP para realizar a agregação de novos dispositivos. O NAP envia pacotes de anúncio que são recebidos por dispositivos que desejam entrar na rede. Através destes links de join, os dispositivos trocam as primeiras mensagens com o gerenciador para iniciar o processo de agregação. Foi determinado que o tamanho deste superframe é 100 timeslots. Este tamanho torna o período dos pacotes de anúncio suficiente para que novos dispositivos escutem a rede e realizem o processo de agregação. Um superframe de anúncio com tamanho menor diminuiria o período dos pacotes de anúncio. Esta estratégia pode ser interessante para os casos onde dispositivos móveis passam pelo alcance da rede e precisam receber o quanto antes um pacote de anúncio para iniciar rapidamente o processo de agregação. Por outro lado, a diminuição deste superframe diminui também a escalabilidade da rede uma vez que mais recursos são utilizados para a divulgação da rede e menos recursos ficam disponíveis para publicação de dados pelos dispositivos.

Figura 19 – Formato do superframe de anúncio e posição dos links.

S0 = 100 timeslots



Fonte: do autor.

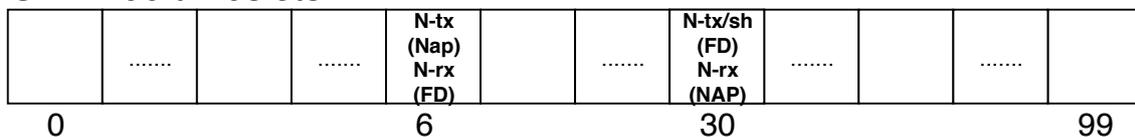
A Figura 20 mostra o superframe de gerenciamento. Os links nos *timeslots* 6 e 9 são utilizados durante o processo de provisionamento, escrevendo-se no NAP um link de transmissão e no dispositivo um link de recepção no *timeslot* 6.

Além disso, o *timeslot* 30 é reservado para que o gateway receba as respostas dos dispositivos. Neste slot, o gateway recebe um link de recepção enquanto que cada dis-

positivo recebe um link de transmissão compartilhado (tx/sh). Um link tx/sh é um link de transmissão onde os dispositivos compartilham um *timeslot* em comum para realizar a comunicação. Para evitar colisões os dispositivos que possuem links deste tipo utilizam o método CSMA/CA para transmitir seus dados no *timeslot* compartilhado. Como considera-se que apenas uma mensagem de gerenciamento pode circular pela rede por vez, os dispositivos podem utilizar o mesmo *timeslot* para responder para o gateway.

Figura 20 – Formato do superframe de gerenciamento.

$S_1 = 100$  timeslots



Fonte: do autor.

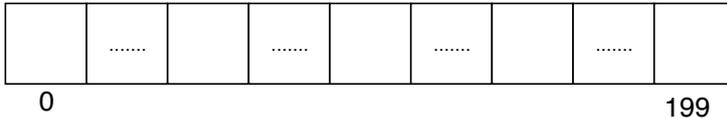
**Superframe de dados:** os superframes de dados são utilizados para os dispositivos enviarem periodicamente seus dados no caso de sensores ou receberem dados do gateway no caso de atuadores. Para cada taxa de transferência de dados do processo pode ser criado um superframe. Como apresentado anteriormente estes superframes devem ser configuradas como múltiplos entre si. No caso do WH estes tempos iniciam com o menor valor, que é 250 ms, e seguem da seguinte forma: 500 ms, 1 s, 2 s, 4 s, 8 s, 16 s, 32 s e 64 s. Após, podem ser definidos como múltiplos de 1 minuto (FOUNDATION, 2008).

Cada período de publicação  $P_i$  suportado pela rede tem um superframe exclusivo. Todos os dispositivos com período  $P_i$  vão ter seus links de dados alocados no mesmo superframe. A Figura 21 apresenta os superframes de dados. O gerenciador gera um superframe  $S_i$  para cada  $P_i$  presente na rede. Os links dos nós com período  $P_i$  devem ser alocados em  $S_i$ . Além dos superframes, o gerenciador armazena uma matriz de escalonamento  $M$  onde as colunas são referentes aos *timeslots* e as linhas referentes ao offset de canais. O tamanho da matriz escalonamento  $M$  é  $N_{canais} \times T$ . A representação como uma matriz é comum na literatura como mostrado anteriormente, porém para o caso de uma rede em estrela, é possível considerar que a matriz de escalonamento seja representada por um vetor, uma vez que não ocorrem duas comunicações em frequências diferentes e em um mesmo *timeslot*. Esta abordagem facilita na imple-

mentação dos algoritmos de escalonamento.

Figura 21 – Formato dos superframes de dados.

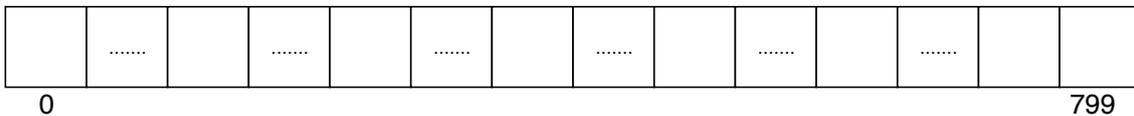
S2 = 200 timeslots



S3 = 400 timeslots



S4 = 800 timeslots



Fonte: do autor.

Porém neste trabalho foram consideradas transmissões de *uplink* (dados saindo dos dispositivos com destino ao gateway) e transmissões de *downlink* (dados saindo do gateway em direção aos dispositivos). Também foram considerados links redundantes, que é uma estratégia comum para aumentar a confiabilidade da rede.

Por simplicidade neste trabalho foi determinado que para todos os dispositivos serão alocados quatro links: transmissão e retransmissão de *uplink* e *downlink*. Em uma aplicação real, diferentes quantidades de *uplink* e *downlink* poderiam ser distribuídos entre os sensores e atuadores. Por exemplo, um atuador não necessariamente precisa publicar dados (*uplink*) periodicamente na mesma frequência com que recebe do controlador (*downlink*), logo a quantidade de *uplink* pode ser diferente da de *downlink*. Além disso, não é considerado o reuso da frequência e por portanto, não podem ocorrer duas transmissões em um mesmo *timeslot*, mesmo que em canais diferentes.

#### 4.2.3 Processo de agregação de um dispositivo de campo na rede

O processo de agregação de um dispositivo na rede precisou ser modificado no gerenciador para que a utilização de múltiplos superframes fosse possível. As principais mudanças ocorreram na função *fd\_formation()*, que como dito anteriormente, é formada por uma estrutura de *switch/case*. Novos *cases* foram adicionados para completar o processo de provisionamento.

Inicialmente o NAP envia pacotes de anúncio, divulgando a rede para que novos dispositivos possam fazer parte. Um dispositivo de campo que esteja previamente configurado com a chave e com o ID da rede pode escutar o meio físico a fim de receber um pacote de anúncio. Assim que o dispositivo recebe um pacote de anúncio, inicia-se o processo de agregação. O dispositivo responde um pacote de anúncio enviando ao gateway os comandos 0, 20 e 787, que são referentes às identificações do próprio dispositivo e dos seus vizinhos. No protocolo WH esta resposta é chamada de *join response*.

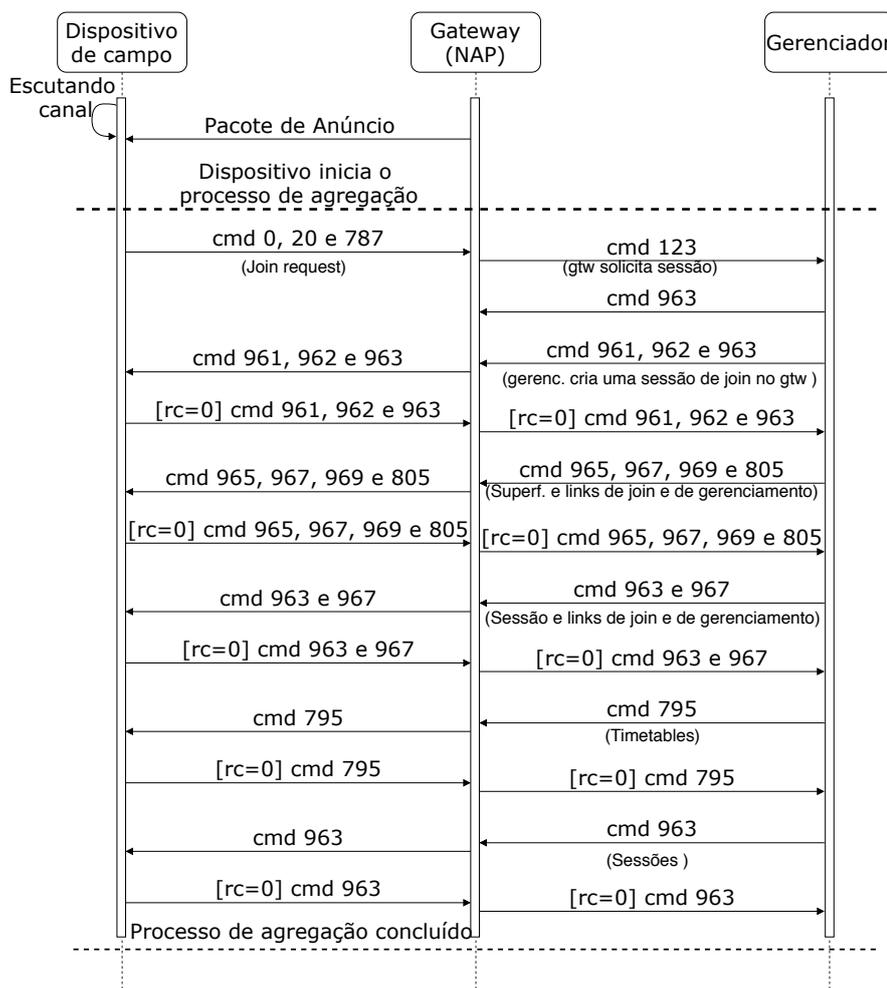
No momento em que o gateway recebe a *join response* ele envia ao gerenciador o comando especial 123. Este é um comando não-público e indica ao gerenciador que foi iniciado o processo de agregação de um novo dispositivo e que o gateway precisa de uma sessão com o dispositivo. O gerenciador, através do comando 963, cria uma sessão entre o gateway e o dispositivo de campo permitindo a troca de diversos comandos entre o dispositivo de campo e o gerenciador através do gateway. Dentre os comandos enviados pelo gerenciador estão os comandos 961, 962 e 963 que são comandos referentes à autenticação do dispositivo na rede. A partir do recebimento destes comandos, o dispositivo passa a ter uma sessão segura para trocar informações com o gerenciador.

A partir de então, são enviados ao dispositivo os superframes e os links de comunicação através dos comandos 965 e 967. Neste momento o dispositivo recebe links provisórios que serão utilizados durante o processo de agregação para o envio dos demais comandos.

O fluxo de mensagens é dado pelo gerenciador enviando comandos para o dispositivo de campo e o dispositivo de campo respondendo estes comandos para o gerenciador. Dentro da resposta está o *response code* que possui o valor zero quando os comandos recebidos são executados corretamente pelo dispositivo. A norma estabelece diferentes códigos de resposta para os casos em que os comandos não são aceitos ou executados corretamente pelos dispositivos. Todas as mensagens possuem ACKs, que foram omitidos nesta figura.

A Figura 22 apresenta um diagrama de sequências que demonstra a troca de mensagens entre o gerenciador de rede e os dispositivos de campo durante o processo de agregação.

Figura 22 – Diagrama de sequência do processo de agregação.



Fonte: do autor.

A Tabela 3 descreve os comandos enviados pelo dispositivo ao gateway após escutar o pacote de anúncio.

Tabela 3 – Comandos enviados pelo dispositivo após escutar o pacote de anúncio.

Comando	Descrição	Informações
0 - Read Unique Identifier	Informações sobre a identificação do dispositivo	Tipo do dispositivo, ID, versão
20 - Read Long Tag	Retorna a Long Tag do dispositivo	Long Tag (32 bytes)
787 - Report Neighbor Signal Levels	Intensidade de sinal de todos os vizinhos	RSL dos vizinhos

A Tabela 4 descreve a função dos comandos utilizados no processo de agregação.

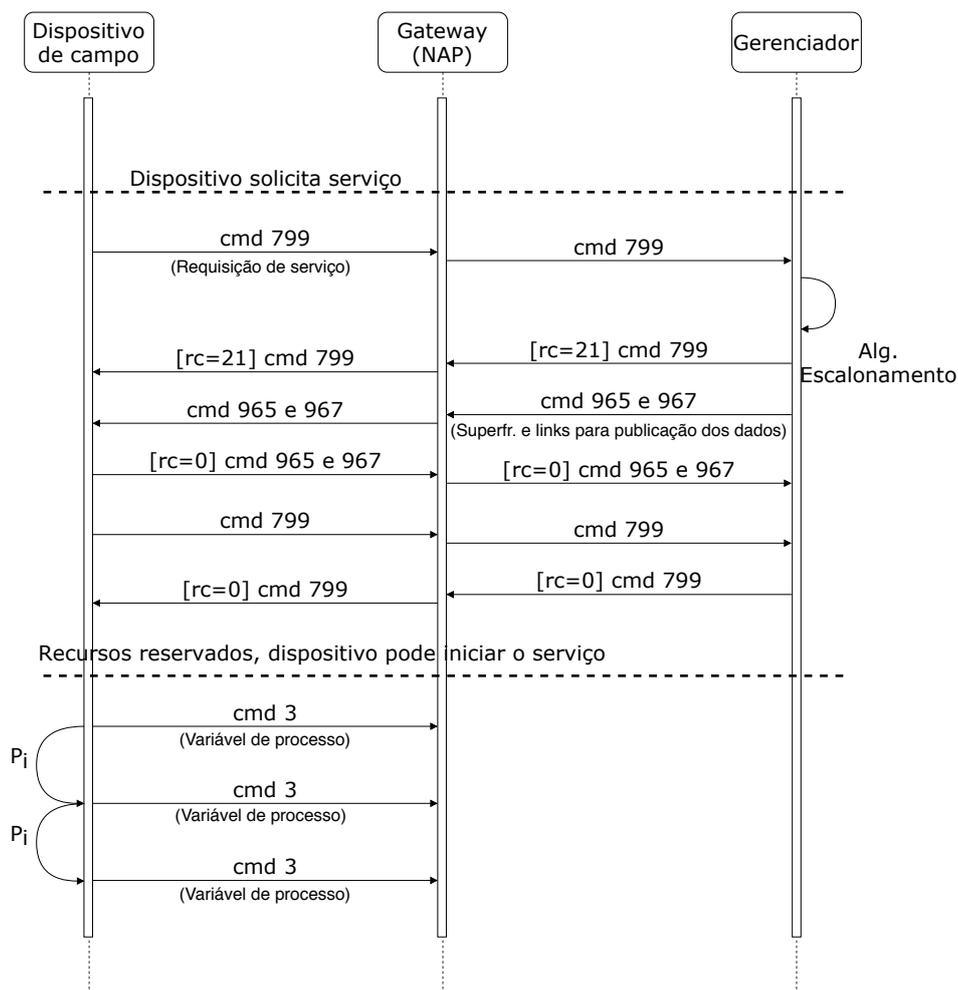
Assim que o processo de agregação é concluído, o dispositivo está pronto para so-

Tabela 4 – Comandos enviados pelo gerenciador ao dispositivo durante o processo de agregação.

Comando	Descrição	Informações
795 - Write Timer Interval	Configura os valores para os temporizadores do dispositivo	Tipo do temporizador, Intervalo de tempo
805 - Write Radio CCA Mode	Escreve o modo de CCA do rádio	Modo CCA: 0 - desabilitado, 1 - habilitado
961 - Write Network Key	Escreve a chave de rede no dispositivo	Valor da chave, ASN para execução do comando
962 - Write Device Nickname Address	Escreve o nickname no dispositivo	Nickname
963 - Write Session	Cria uma sessão de comunicação	Tipo de sessão, Nick do peer, ID do peer, ASN de execução do comando
965 - Write Superframe	Escreve superframe no dispositivo.	Superframe ID, Tamanho do Superframe, ASN de execução do comando
967 - Write Link	Escreve um link no dispositivo	Superframe ID, <i>timeslot</i> , offset de canal, opção e tipo
969 - Write Graph	Adiciona uma conexão a um dispositivo	ID do Grafo, Nickname do vizinho

licitar um serviço ao gerenciador para publicação das suas variáveis de processo. O serviço é solicitado através do comando 799. Um dos parâmetros que estão contidos no comando 799 é o período da variável de processo. Após receber a solicitação de serviço, o gerenciador executa o algoritmo de escalonamento, e durante este processo envia ao dispositivo de campo uma resposta do comando 799 com o *response code* 21. Este valor indica ao dispositivo de campo que o mesmo deve aguardar enquanto o gerenciador verifica se existem recursos disponíveis para a aceitação do serviço. Após a execução do algoritmo de escalonamento, o gerenciador envia ao dispositivo o superframe e os links calculados pelo algoritmo. O dispositivo, mais uma vez, envia o comando 799 solicitando serviço, e desta vez o gerenciador responde com o código 0, indicando que o dispositivo pode iniciar a publicação dos seus dados. A Figura 23 apresenta um diagrama de sequência do processo de requisição de serviço.

Figura 23 – Diagrama de sequência do processo de solicitação de serviço.



Fonte: do autor.

A Tabela 5 descreve a função dos comandos utilizados na solicitação de serviço. O comando 3 traz informações das variáveis de processo do dispositivo, informando valores de até quatro variáveis.

Tabela 5 – Comandos enviados ao gerenciador pelo dispositivo para solicitação de serviço.

Comando	Descrição	Informações
3 - Read Dynamic Variables	Leitura das variáveis dinâmicas do dispositivo	Variável primária, secundária, terciária e quaternária
799 - Request Service	Dispositivo realiza uma requisição de serviço	ID do serviço, nickname do peer, Período de publicação

#### 4.2.4 Algoritmos de escalonamento

Para verificar se o tempo de escalonamento pode influenciar no funcionamento de RSFI, foram implementadas algumas estratégias de escalonamento de links para determinados cenários. Como o intuito é verificar o tempo de execução dos algoritmos de escalonamento foram realizados testes simulados onde são escalonados apenas os links dos superframes de dados. A seguir são descritas as estratégias implementadas.

- **Algoritmo de escalonamento DM e EDF:**

As estratégias DM e EDF são técnicas clássicas de escalonamento de tarefas que utilizam prioridades para realizar a alocação. No caso de uma RSFI, são alocados links para as transmissões conforme a prioridade dos pacotes gerados pelos fluxos de um conjunto de dispositivos. As prioridades são baseadas nos *deadlines* dos pacotes. A partir do conjunto de dispositivos é gerado o conjunto de links  $L$ . A cada iteração do algoritmo o conjunto  $L$  é ordenado e o primeiro elemento do conjunto é escolhido para ser alocado.

Estes algoritmos consideram o escalonamento de tarefas e, neste trabalho, as tarefas representam os links de comunicação dos dispositivos. Cada uma das tarefas é referente a um dos links que serão escalonados. Cada tarefa que representa um link  $l_{ij}$  de um dispositivo  $i$  é constituída por:

$$T_i = (P_i, D_i, R_i)$$

onde:

- $P$ : período no qual o dispositivo  $i$  publica seus dados;
- $D$ : *deadline* no qual o dado publicado por  $i$  deve chegar ao destino;
- $R$ : release time, instante em que a tarefa  $T_i$  estará pronta para ser executada.

Para evitar que o link de retransmissão  $l_{i1}$  seja alocado no *timeslot* seguinte ao link de transmissão  $l_{i0}$ , foi estabelecido que as tarefas referentes aos links de um determinado dispositivo tenham release times diferentes. O release time da primeira tarefa é definido como 0, ou seja, para todos os dispositivos a primeira tarefa esta pronta para ser executada no *timeslot* 0. O release time das demais tarefas são determinados por  $r_{i,j+1} = S_{l_{ij}} + 4$  sendo  $S_{l_{ij}}$  o slot onde  $l_{ij}$  foi alocado. Com isso, os links de um dispositivo é escalonado com 4 *timeslots* de intervalo entre eles.

Todas as mensagens devem chegar ao seu destino respeitando os seus respectivos *deadlines*. Para isso, os links de comunicação devem ser agendados de forma que, mesmo que ocorra uma falha no link de transmissão, o link de retransmissão atenda o *deadline*. No escopo deste trabalho os deadlines são definidos aleatoriamente, sendo que o *deadline*  $D_i$  para um nó  $i$  é dado por  $h * P_i$  sendo  $h = [0.5, 1]$ .

*Deadline monotonic*: o DM é um algoritmo de escalonamento de prioridade fixa que define as prioridades a partir do *deadline* relativo dos pacotes e portanto, é possível alocar os links nas mesmas posições para todos os ciclos do respectivo superframe. O Algoritmo 1 apresenta as operações do DM.

---

**Algoritmo 1:** Algoritmo de Escalonamento DM

---

**Entrada:**  $V$  // Conjunto de dispositivos  
**Saída:**  $M$  // Matriz Escalonamento  
 $L$  // Lista de links  
 $S$  // Lista de superframes

```

1 início
2   lt ← lista<Tarefas>; // Cria uma lista do tipo tarefas
3   para todo  $v \in V$  faça
4     tf ← nova tarefa; // Cria uma tarefa que representa um link de  $v$ 
5     lt ← tf; // Adiciona tarefa na lista de tarefas
6     se  $S_i$  não existe então
7        $S_i$  ← novo spf com tamanho  $P_i$ ; //  $S_i$  é o spf p/ período  $P_i$  de  $v$ 
8        $S \leftarrow S_i$  // Adiciona  $S_i$  a  $S$ 
9     fim
10  fim
11  sort(lt); // Ordena a lista de tarefas por ordem de deadline
12   $slot_t \leftarrow 0$ ; // Inicia no timeslot 0 de  $M$ 
13  enquanto  $lt \neq \emptyset$  faça
14    t ← lt[0]; // Pega primeira tarefa da lista
15     $m \leftarrow \frac{P_{max}}{P_i}$ ; // Verifica qts vezes  $S_i$  repete em  $M$ 
16    aloca o link referente a t uma vez em  $S_i$  no  $slot_t$  e  $m$  vezes em  $M$ ;
17    delete(t); // retira tarefa da lista;
18    se quantidade de links alocados para  $v < 4$  então
19      tf ← nova tarefa; // Adiciona tarefa que representa o próx. link de  $v$ 
20      lt ← tf; // Adiciona tarefa na lista de tarefas
21      sort(lt); // Reordena a lista de tarefas
22    fim
23     $slot_t + +$ ; // Incrementa timeslot;
24  fim
25  se Não conseguiu alocar todos os links então
26    retorna FAIL;
27  fim
28 fim
```

---

*Earliest Deadline First*: o EDF é um algoritmo de prioridade dinâmica que considera o *deadline* absoluto dos pacotes para definir as prioridades. Desta forma as transmissões de um mesmo fluxo pode possuir diferentes prioridades nos ciclos do superframe, consequentemente os links serão alocados em *timeslots* diferentes de um ciclo para outro. Isso torna inviável a utilização de um superframe para cada período. A partir disso, é gerado um superframe de tamanho  $T$  para todos os dispositivos onde cada um receberá um número diferente de links conforme a sua taxa de publicação. Esta estratégia apesar de diminuir o número de superframes presentes na rede faz com que mais transmissões de RF ocorram para que o gerenciador escreva nos dispositivos os links de comunicação. O Algoritmo 2 apresenta a sequência de operações do EDF.

---

**Algoritmo 2:** Algoritmo de Escalonamento EDF
 

---

**Entrada:**  $V$  // Conjunto de dispositivos  
**Saída:**  $M$  // Matriz Escalonamento  
 $L$  // Lista de links  
 $S$  // Lista de superframes

```

1 início
2    $lt \leftarrow \text{lista}\langle \text{Tarefas} \rangle$ ; // Cria uma lista do tipo tarefas
3    $S_{max} \leftarrow \text{spf com tamanho } P_{max}$ ; //  $S_{max}$  é o spf com tamanho do maior
   período  $P_i$ 
4   para todo  $v \in V$  faça
5      $tf \leftarrow \text{nova tarefa}$ ; // Cria uma tarefa que representa um link de  $v$ 
6      $lt \leftarrow tf$ ; // Adiciona tarefa na lista de tarefas
7   fim
8    $\text{sort}(lt)$ ; // Ordena a lista de tarefas por ordem de deadline
9    $slot_t \leftarrow 0$ ; // Inicia no timeslot 0 de  $M$ 
10  enquanto  $lt \neq \emptyset$  faça
11     $t \leftarrow lt[0]$ ; // Pega primeira tarefa da lista
12    aloca o link referente a  $t$  em  $S_{max}$  no  $slot_t$  e em  $M$ ;
13     $\text{delete}(t)$ ; // retira tarefa da lista;
14    se quantidade de links alocados para  $v < (4 * \frac{P_{max}}{P_i})$  então
15       $tf \leftarrow \text{nova tarefa}$ ; // Adiciona tarefa que representa o próx. link de  $v$ 
16       $lt \leftarrow tf$ ; // Adiciona tarefa na lista de tarefas
17       $\text{sort}(lt)$ ; // Reordena a lista de tarefas
18    fim
19     $slot_t + +$ ; // Incrementa timeslot;
20  fim
21  se Não conseguiu alocar todos os links então
22    retorna FAIL;
23  fim
24 fim

```

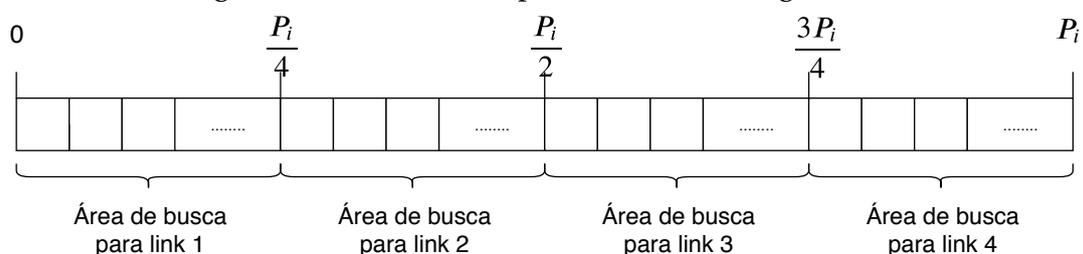
---

• **Algoritmo de escalonamento Han:**

O trabalho de (HAN *et al.*, 2011) apresenta um algoritmo de escalonamento para RSFI com topologia em malha. O algoritmo é composto por duas etapas. A primeira delas é responsável por determinar qual dispositivo terá seus links alocados e em que região do superframe será feita a busca por *timeslots* vagos. A segunda etapa leva em consideração questões de roteamento como caminho e rota redundante. Neste trabalho, é utilizada apenas a primeira parte do algoritmo, onde é realizada a busca pelos *timeslots* onde serão alocados os links.

O superframe é dividido em duas fases. A primeira é chamada de fase de sensoriamento onde os dispositivos publicam suas variáveis de processo para o gateway. A segunda é a fase de controle, e nesta fase, o NM gera mensagens de controle e envia elas aos dispositivos. O algoritmo aplica a política de Fastest Sample Rate First (FSRF) para realizar o escalonamento, ou seja, os dispositivos que possuem maior taxa de atualização são escalonados primeiro. A partir de um conjunto de dispositivos o algoritmo realiza a alocação de quatro links para cada dispositivo: dois links para a fase de sensoriamento e dois para a fase de controle. Originalmente, o algoritmo de Han foi proposto para uma topologia em malha, porém os conceitos podem ser aplicados a topologia estrela utilizada neste trabalho. A Figura 24 apresenta um exemplo de superframe com tamanho  $P_i$  onde  $F_i$  é o fluxo do dispositivo  $i$  e  $P_i$  é o período em que o fluxo  $F_i$  ocorre. O superframe é dividido em quatro partes. Para a alocação do primeiro link, o algoritmo realiza a busca por um slot vago no intervalo 0 a  $\frac{P_i}{4}$ , para o segundo link no intervalo de  $\frac{P_i}{4}$  à  $\frac{P_i}{2}$  e assim sucessivamente.

Figura 24 – Divisão do superframe na estratégia de Han.



Fonte: do autor.

Cada um dos links é alocado individualmente. No caso deste trabalho, são alocados 4 links, desta forma o algoritmo faz quatro buscas independentes dentro da matriz de escalonamento para encontrar uma posição disponível. O Algoritmo 3 apresenta as

etapas da estratégia apresentada por Han para o escalonamento dos dispositivos para uma topologia estrela.

---

**Algoritmo 3:** Algoritmo de Escalonamento Han

---

**Entrada:**  $V$  // Conjunto de dispositivos  
**Saída:**  $M$  // Matriz de escalonamento  
 $L$  // Lista de links  
 $S$  // Lista de superframes

```

1 início
2   Ordena  $V$  em ordem crescente de  $P : P_1 < P_2 < \dots < P_k$ ;
3   para  $P_1$  até  $P_k$  faça
4     Gerar o superframe de dados  $S_i$ ;
5     para todo nó  $v \in V_i$  faça
6        $a = 0$ ;
7        $b = \frac{P_i}{4}$ ;
8       para cada Link  $l_{ij}$  de  $v$  faça
9         Procurar um slot  $t_i$  disponível entre  $a$  e  $b$  em  $M$ ;
10        Escalonar o link  $l_{ij}$  no slot  $t_i$ ;
11        Alocar o slot  $t_i$  em  $M$  e em  $S_i$ ;
12         $a = a + \frac{P_i}{4}$ ;
13         $b = b + \frac{P_i}{4}$ ;
14      fim
15      se Não conseguiu alocar todos os links então
16        retorna FAIL;
17      fim
18    fim
19  fim
20 fim

```

---

• **Algoritmo de escalonamento Han-MO:**

Foi realizada uma adaptação do algoritmo de Han utilizando o conceito de macro operações. Este conceito consiste em agrupar um conjunto de tarefas e realizar a alocação deste conjunto ao invés de alocar cada tarefa individualmente. O objetivo é reduzir o número de ciclos necessários para executar o processo de escalonamento, alocando todos os links de um dispositivo  $v_i$  simultaneamente na matriz  $M$ . Esta estratégia foi adaptada para o escalonamento de links em RSFI. Considerando-se que são conhecidos os períodos de publicação de dados que a rede suporta, define-se como uma macro operação um bloco de  $t$  timeslots onde  $t$  é igual ao período de publicação do dispositivo. É realizada a busca por uma posição em  $M$  que comporte  $t$ . Assim que o algoritmo encontra esta posição, ele aloca os quatro links do dispositivo de uma só

vez.

A Figura 25 apresenta um exemplo de vetor ocupação e da macro operação da figura anterior. Como mostra a figura, o algoritmo percorre o vetor escalonamento tentando “encaixar” a macro operação. Quando uma posição é encontrada, todos os links são alocados de uma só vez.

O Algoritmo 4 apresenta a estratégia de escalonamento Han-MO para o escalonamento de links em redes industriais sem fio.

---

**Algoritmo 4:** Algoritmo de Escalonamento Han-MO

---

**Entrada:**  $V$  // Conjunto de dispositivos  
**Saída:**  $M$  // Matriz Escalonamento  
 $L$  // Lista de links  
 $S$  // Lista de superframes

```

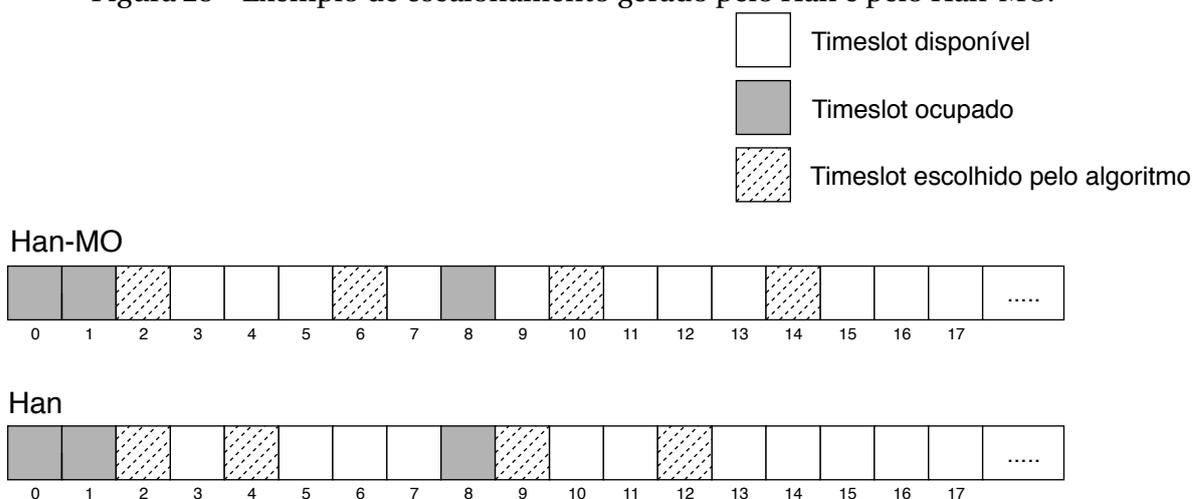
1 início
2   Ordena  $N$  em ordem crescente de  $P : P_1 < P_2 < \dots < P_k$ ;
3   para  $P_1$  até  $P_k$  faça
4     Gerar o superframe de dados  $S_i$ ;
5     para todo  $v \in V$  faça
6        $MO$  // Cria a macro operação;
7       Procurar uma posição em  $M$  onde  $MO$  possa ser encaixada;
8       Escalonar os 4 links nos timeslots referentes a posição encontrada;
9       se Não conseguiu alocar todos os links então
10        retorna FAIL;
11      fim
12    fim
13  fim
14 fim

```

---

A diferença entre os métodos Han e Han-MO consiste na forma como os algoritmos buscam slots para alocação dos links. Considerando a alocação de quatro links para cada dispositivo o Han aloca os 4 links individualmente, já o Han-MO busca alocar os 4 links simultaneamente. A Figura 25 apresenta a diferença entre o escalonamento gerado pelos dois métodos em um caso onde um dispositivo que publica a cada 16 *timeslots* tem seus 4 links alocados pelos dois algoritmos. A Figura apresenta dois vetores que representam os *timeslots* ocupados e disponíveis na rede. Os *timeslots* cinzas representam links que já estão alocados no superframe. Os *timeslots* brancos representam *timeslots* disponíveis para receber links. Já, os *timeslots* tracejados representam os slots escolhidos pelos algoritmos para alocar os links. Percebe-se que o Han-MO mantém uma distância fixa entre os links.

Figura 25 – Exemplo de escalonamento gerado pelo Han e pelo Han-MO.



Fonte: do autor.

#### 4.2.5 Proposta de algoritmo de escalonamento

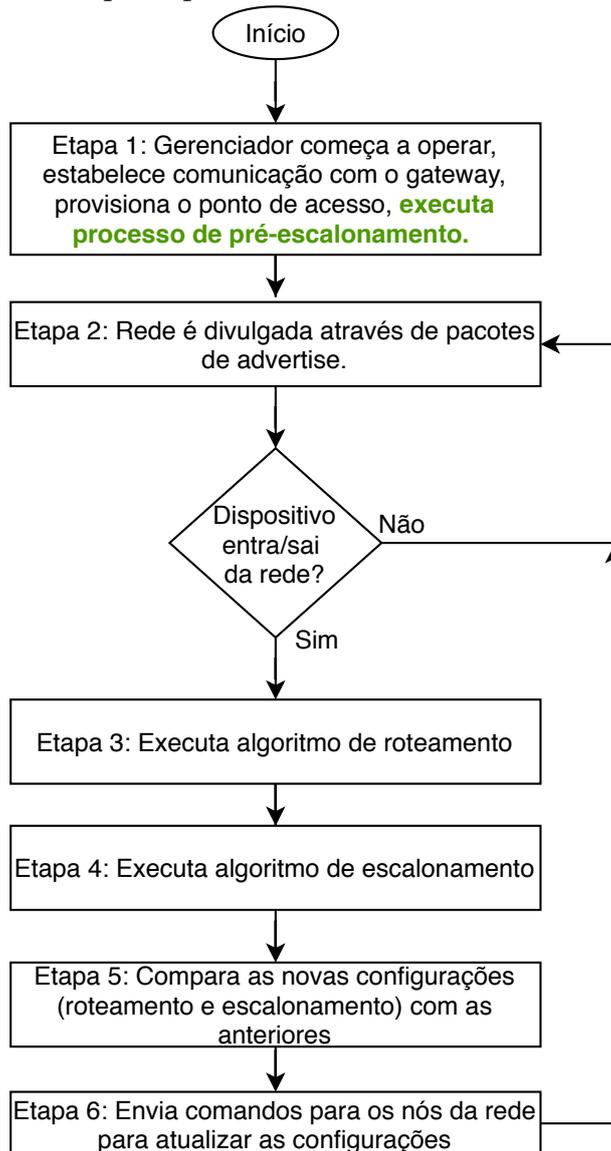
A partir de estudos realizados das técnicas de escalonamento de links mais relevantes presentes na literatura, e do entendimento de que o tempo de execução das tarefas de gerenciamento, em especial do algoritmo de escalonamento pode ser um fator relevante na implementação de RSFI, este trabalho apresenta uma proposta de algoritmo de escalonamento que visa reduzir o tempo de execução do processo de escalonamento e ao mesmo tempo trabalha com múltiplos superframes.

A técnica chamada de escalonamento de links utilizando estruturas pré-alocadas (do inglês Link Scheduling using Pre-Allocated structures ou LSPA) consiste em duas fases: a primeira chamada de pré-escalonamento é executada no momento em que o gerenciador começa a operar, a segunda chamada de escalonamento é executada durante o funcionamento da rede e utiliza informações geradas na primeira etapa. A Figura 26 apresenta uma possível rotina do gerenciador de rede em uma RSFI da mesma forma como mostrado anteriormente. É possível perceber que o processo de pré-escalonamento foi inserido na Etapa 1. Ambas as fases são descritas abaixo.

##### • Pré-escalonamento:

Este processo leva em consideração o número de links que serão alocados para cada dispositivo e também os períodos de publicação suportados pela rede. Desta forma, na Etapa 1 da Figura 26, o gerenciador gera uma lista de estruturas que representam os *timeslots* onde os dispositivos serão alocados. Estas estruturas consistem em um grupo de  $k$  *timeslots* onde  $k$  é a quantidade de links que serão alocados

Figura 26 – Inserção da etapa de pré-escalonamento na rotina do gerenciador de rede.

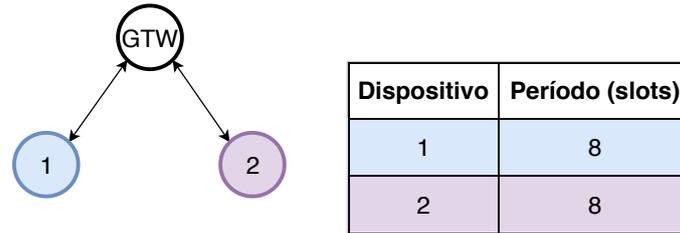


Fonte: do autor.

para cada dispositivo (neste trabalho  $k = 4$ ). Uma vez que o gerenciador tem conhecimento dos possíveis períodos de publicação dos dispositivos e também da quantidade de links que serão alocados para cada um, é possível prever a quantidade de dispositivos que poderão ser alocados.

A Figura 27 apresenta um exemplo: considerando uma rede com 2 dispositivos em que ambos publicam seus dados a cada 8 *timeslots* poderão ser alocados apenas dois dispositivos, pois cada dispositivo terá 4 *timeslots* para uso exclusivo. Desta maneira, o gerenciador de rede consegue prever quais são os *timeslots* em que os links de um determinado dispositivo poderão ser alocados.

Figura 27 – Exemplo de rede com dois dispositivos que publicam a cada 8 *timeslots*.

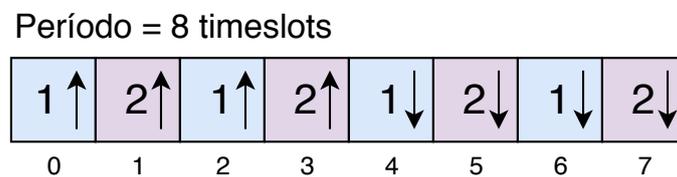


2 links de uplink  
2 links de downlink

Fonte: do autor.

A Figura 28 mostra um possível escalonamento para este exemplo. Os quatro links de cada dispositivo estão igualmente espaçados dentro do superframe. Os números indicam qual dispositivo está ocupando o *timeslot*. As setas indicam se o link é de *uplink* (↑) ou *downlink* (↓).

Figura 28 – Dispositivos 1 e 2 escalonados no superframe  $S_i$  de tamanho 8



Fonte: do autor.

Apesar de ser possível prever quais *timeslots* um dispositivo de um determinado período vai utilizar, não se sabe quantos dispositivos de cada período vão entrar na rede e nem a ordem em que vão entrar, por isso não é possível realizar o pré-escalonamento dos links dos dispositivos. Desse modo, a pré-alocação das estruturas é uma maneira de reservar *timeslots* sem necessariamente saber quais links vão ocupá-lo. Para cada período de publicação que a rede atende, é gerado um conjunto  $C_i$  de estruturas. Cada conjunto  $C_i$  possui  $\frac{P_i}{4}$  estruturas denominadas como  $C_{i_n}$  onde  $n$  é definido por (3).

$$0 < n < \left(\frac{P_i}{4}\right) - 1; \quad (3)$$

Cada estrutura  $C_{i_n}$  possui  $j$  *timeslots* calculados por (4)

$$timeslots\ de\ C_{i_n} = j * \frac{P_i}{4} + n; \quad (4)$$

Onde:

$$0 < j < 3;$$

Para exemplificar, a Figura 29 apresenta a pré-alocação das estruturas para períodos de 8 e 16 *timeslots*. O gerenciador gera estes vetores durante o processo de pré-escalamento que ocorre na Etapa 1. Em vez de atribuir links aos *timeslots*, em um primeiro momento são atribuídos os índices de cada estrutura e estas informações são posteriormente utilizadas no processo de escalamento (Etapa 4).

Figura 29 – Superframes com as estruturas dos períodos 8 e 16 pré-alocadas.

$P_i = 8$

C8 <sub>0</sub>	C8 <sub>1</sub>						
0	1	2	3	4	5	6	7

$P_i = 16$

C16 <sub>0</sub>	C16 <sub>1</sub>	C16 <sub>2</sub>	C16 <sub>3</sub>	C16 <sub>0</sub>	C16 <sub>1</sub>	C16 <sub>2</sub>	C16 <sub>3</sub>	C16 <sub>0</sub>	C16 <sub>1</sub>	C16 <sub>2</sub>	C16 <sub>3</sub>	C16 <sub>0</sub>	C16 <sub>1</sub>	C16 <sub>2</sub>	C16 <sub>3</sub>
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

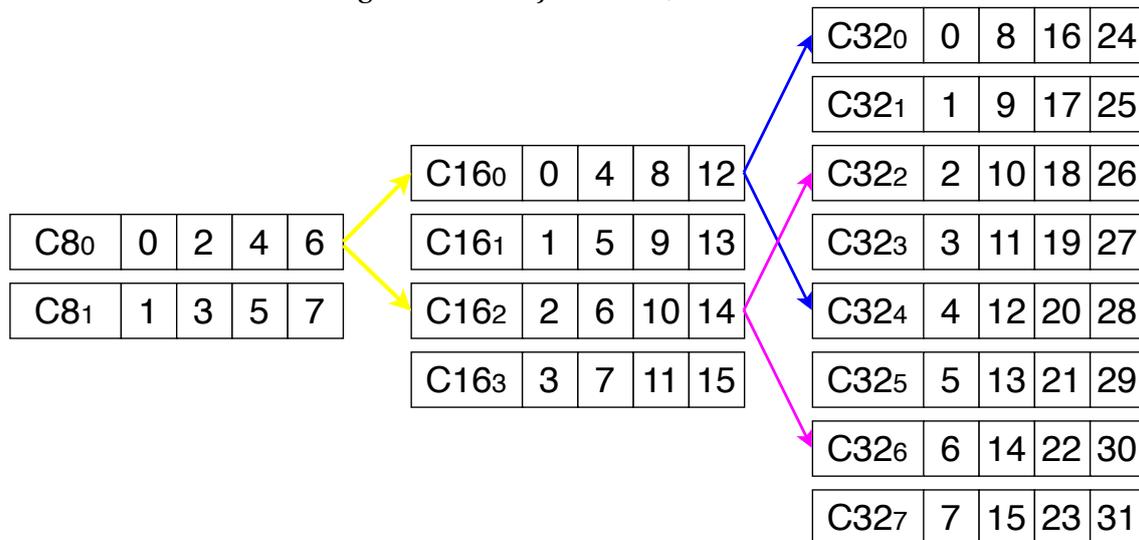
Fonte: do autor. .

Algumas das estruturas tem *timeslots* conflitantes, conseqüentemente no momento em que uma estrutura é utilizada todas as demais que conflitem com a mesma devem ser desabilitadas. A Figura 30 traz um exemplo com os conjuntos de estruturas para períodos de publicação de 8, 16 e 32 *timeslots*. As estruturas são representadas por um índice do  $C_{i_n}$  e por quatro valores que representam os slots reservados para aquela estrutura.

É possível perceber a relação entre os conjuntos de estruturas como por exemplo, a estrutura  $C_{8_0}$  tem *timeslots* conflitantes com as estruturas  $C_{16_0}$ ,  $C_{16_2}$ ,  $C_{32_0}$ ,  $C_{32_2}$ ,  $C_{32_4}$  e  $C_{32_6}$ . Desta forma, é importante manter o controle adequado das estruturas ocupadas e disponíveis para evitar que uma mesma estrutura seja utilizada duas vezes causando conflitos entre os links. Neste exemplo, caso a estrutura  $C_{8_0}$  seja utilizada, as demais estruturas mencionadas devem ser desabilitadas temporariamente.

As figuras apresentadas anteriormente exemplificam o conceito proposto pelo LSPA,

Figura 30 – Conjuntos C8, C16 e C32.



Fonte: do autor.

porém os períodos utilizados neste trabalho não são os mesmos apresentados acima. Para que fosse possível realizar a implementação dos algoritmos no gerenciador de rede, foram utilizados alguns dos períodos suportados pelo protocolo WH como: 250 ms, 500 ms, 1 s, 2 s, 4 s, 8 s e 16 s. Os conjuntos gerados bem como a quantidade de estruturas de cada conjunto são apresentados na Tabela 6.

Tabela 6 – Conjuntos de estruturas gerados.

Conjunto	Quantidade
C25	6
C50	12
C100	25
C200	50
C400	100
C800	200
C1600	400

O Algoritmo 5 apresenta a sequência de operações realizadas para gerar os conjuntos de estruturas. Este algoritmo é executado no momento da inicialização do gerenciador de rede e recebe como entrada os períodos de publicação que a rede suporta e

a quantidade de links que cada dispositivo vai receber.

---

**Algoritmo 5:** Algoritmo para gerar estruturas de escalonamento.

---

**Entrada:**  $P$  // Períodos suportados pela rede em *timeslots*

$k$  // nº de links que cada dispositivo recebe

**Saída:**  $C$  // Matriz com todas as estruturas geradas (conjuntos de estruturas)

```

1 início
2   q ← 0;
3   para cada período  $P_i$  faça
4     t ←  $\frac{P_i}{k}$  // total de estruturas para cada período
5     para  $i = 0$  até  $i = t$  faça
6       C[q][0] ←  $P_i * 10 + i$  // id da estrutura;
7       C[q][1] ←  $i$  // slot p/ 1º link;
8       C[q][2] ←  $\frac{P_i}{4} + i$  // slot p/ 2º link;
9       C[q][3] ←  $\frac{P_i}{2} + i$  // slot p/ 3º link;
10      C[q][4] ←  $3\frac{P_i}{4} + i$  // slot p/ 4º link;
11      q++;
12    fim
13  fim
14 fim

```

---

• **Escalonamento:**

O processo de escalonamento consiste na busca por estruturas pré-alocadas. Para cada dispositivo  $v$  o algoritmo realiza a busca por uma estrutura  $Ci_n$  disponível que atenda ao período de publicação  $P_i$  do dispositivo. O Algoritmo 6 apresenta o funcionamento deste processo.

Para verificar se uma determinada estrutura conflita com alguma já utilizada é mantida uma matriz de conflitos. Esta matriz possui o número de linhas igual ao tamanho do menor conjunto de estruturas. Neste trabalho o menor conjunto considerado foi o C25 que possui seis estruturas e portanto a matriz possui seis linhas. O comprimento desta matriz foi arbitrariamente definido como 10.

No momento em que o algoritmo encontra a estrutura disponível para o determinado período (linha 10) já é possível saber os quatro *timeslots* aos quais serão alocados os links do dispositivo. Desta maneira é feita a alocação dos links na matriz  $M$  utilizando os *timeslots* da estrutura encontrada. A partir de então, a estrutura passa a ser indisponível.

---

**Algoritmo 6:** Escalonamento utilizando estruturas pré-alocadas.
 

---

**Entrada:**  $V$  // Conjunto de dispositivos  
 $C$  // Conjunto de estruturas  
 $MC$  // Matriz de conflitos  
**Saída:**  $M$  // Matriz de escalonamento

```

1 início
2   para cada  $v \in V$  faça
3     //  $i$  = período de  $v$ 
4     para cada  $Ci_n \in Ci$  faça
5       // Avalia cada estrutura candidata pertencente ao conjunto  $Ci$ 
6       se  $Ci_n == indisponível$  então
7         continue;
8       senão
9          $k \leftarrow$  linha em  $MC$  a qual  $Ci_n$  corresponde;
10        para cada elemento  $l \in MC$  na linha  $k$  faça
11          se  $Ci_n$  conflitar com algum elemento em  $MC$  então
12             $Ci_n \leftarrow$  indisponível;
13            break;
14          senão
15            // Encontrou uma estrutura disponível.
16            Aloca os links de  $v$  em  $M$  e em  $S_i$  utilizando os timeslots
17            de  $Ci_n$ ;
18             $Ci_n \leftarrow$  indisponível;  $MC \leftarrow Ci_n$  // Armazena o índice da
19            estrutura  $Ci_n$  na matriz de conflitos
20          fim
21        fim
22      fim
23    fim
  
```

---

Para exemplificar o funcionamento da matriz de conflitos (MC) podemos utilizar um exemplo: quatro dispositivos ( $v_1$ ,  $v_2$ ,  $v_3$  e  $v_4$ ) tem respectivamente os períodos 500 ms, 500ms, 1 s e 2 s e serão escalonados utilizando o algoritmo LSPA. Inicialmente o algoritmo procura uma estrutura disponível para o dispositivo  $v_1$  encontrando a estrutura  $C50_0$  uma vez que até então nenhum dispositivo vou escalonado. O algoritmo utiliza então os *timeslots* da estrutura  $C50_0$  para alocar os links do dispositivo  $v_1$ . Esta estrutura se torna indisponível e seu índice é armazenado na matriz de conflitos (MC) na linha referente a sua correspondente que é a estrutura  $C25_0$ . Na sequência o dispositivo  $v_2$  é escalonado, recebendo os links da estrutura  $C50_1$  que é a próxima disponível para o período de 500 ms. O índice de  $C50_1$  também é armazenado em  $MC$ .

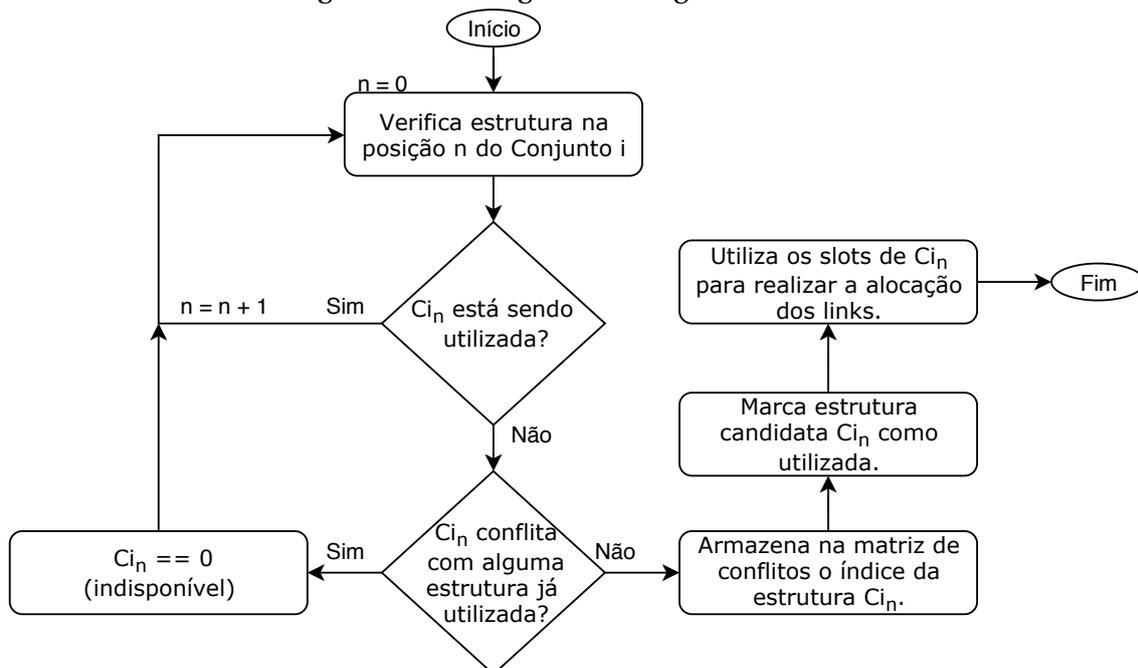
Durante a busca por uma estrutura para o  $v_3$ , o algoritmo começa avaliando a es-

estrutura  $C100_0$ , esta estrutura está disponível, pois não está sendo utilizada por nenhum dispositivo que tem período 1s, porém no momento da verificação da matriz de conflitos, o algoritmo verifica que esta estrutura conflita com a  $C50_0$  e desta forma não deve ser utilizada. O mesmo ocorre com a estrutura  $C100_1$  que conflita com a  $C50_1$ . Portanto o dispositivo  $\nu_3$  acaba recebendo os links da estrutura  $C100_2$ . O mesmo ocorre com o dispositivo  $\nu_4$  que recebe os links da estrutura  $C200_3$ . A Table 7 apresenta a matriz de conflitos  $MC$  resultante após o processo de escalonamento para o exemplo descrito e a Figura 31 apresenta um fluxograma do algoritmo LSPA.

Tabela 7 – Matriz de conflitos  $MC$  resultante após o processo de escalonamento de quatro dispositivos com períodos 500 ms, 500ms, 1 s e 2 s.

Correspondente	Estruturas utilizadas							
$C25_0$	$C50_0$							
$C25_1$	$C50_1$							
$C25_2$	$C100_2$							
$C25_3$	$C200_3$							
$C25_4$								
$C25_5$								

Figura 31 – Fluxograma do algoritmo LSPA.



Fonte: do autor.

## 5 ESTUDOS DE CASO

Para realizar a comparação entre o algoritmo proposto e os demais algoritmos de escalonamento foram propostos alguns estudos de caso. Os testes dos estudos de caso A e B foram realizados de forma *offline*, ou seja, sem a operação de uma rede real. Para isso, foi utilizado o mesmo código do gerenciador de rede construído em linguagem C e foi simulado o escalonamento dos dispositivos de campo. Nestes estudos foram avaliados o tempo de execução dos algoritmos de escalonamento bem como a taxa de sucesso de cada um que representa a quantidade de vezes em que os algoritmos conseguiram alocar um novo dispositivo na rede. Estes estudos tem o propósito de analisar a etapa 4 da Figura 26.

Um terceiro estudo de caso foi proposto com o objetivo de comparar as políticas de escalonamento utilizadas pelo gerenciador de rede proposto e pelo gateway comercial Emerson 1420A. O Gateway comercial utiliza um único superframe para as transmissões de dados, enquanto que o gerenciador proposto faz uso de múltiplos superframes. Nesta análise é avaliada a etapa 6 da Figura 26.

Para comparar os algoritmos de escalonamento, foram propostos estudos de caso, os quais são apresentados a seguir. O estudo de caso A foi subdividido em três casos, sendo chamados de caso A1, A2 e A3 e são referentes à reconfiguração completa da rede, ou seja, o escalonamento parte do zero. Para o estudo de caso B, é considerada uma reconfiguração parcial, isto é, um único dispositivo tenta entrar em uma rede já estabelecida. Para cada caso é variada a quantidade de dispositivos e a taxa de atualização. O número de dispositivo que podem ser escalonados depende da quantidade de links que cada um vai receber e também dos períodos de publicação. A seguinte equação deve ser respeitada durante o escalonamento:

$$k * \left( \frac{n_1}{P_1} + \frac{n_2}{P_2} + \dots + \frac{n_i}{P_i} \right) \leq 1 \quad (5)$$

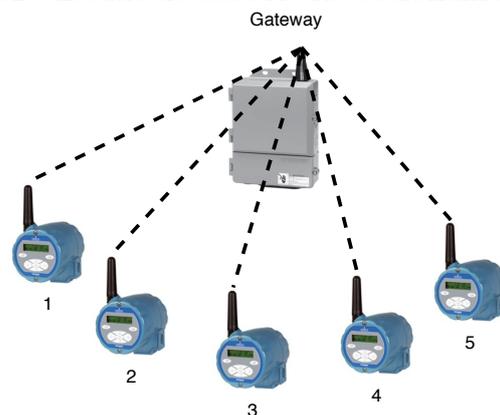
onde:

- $k$  é a quantidade de links que cada dispositivo recebe;
- $n_i$  é a quantidade de dispositivos de um determinado período;
- $P_i$  é o período de publicação dos dispositivos em *timeslots*.

## 5.1 Estudo de Caso A: Escalonamento completo

O caso A considera o reescalamento de todos os dispositivos da rede. É considerada uma rede estrela, como mostra a Figura 32, onde todos os dispositivos serão escalonados com o mesmo número de links ( $k = 4$ ). Neste caso, são avaliados apenas os tempos de execução dos algoritmos de escalonamento, uma vez que todos os algoritmos realizam a alocação de uma mesma quantidade de links e as quantidades de dispositivos seguiram a equação (5).

Figura 32 – Estudo de Caso A - Reescalamento completo da rede.



Fonte: do autor.

### 5.1.1 Caso A1:

No primeiro caso foi considerado o reescalamento completo da rede. Neste cenário o gerenciador executa o processo de escalonamento para todos os dispositivos presentes na rede. Foram analisados 5 grupos de dispositivos (A-E), onde todos os dis-

positivos possuem o mesmo período de publicação. Cada grupo possui uma quantidade diferente de dispositivos. Para cada grupo de dispositivos executou-se 100 vezes cada algoritmo. As quantidades de dispositivos de cada grupo são apresentadas na Tabela 8.

Tabela 8 – Parâmetros do Caso A1.

	Nº de dispositivos				
$P_i$ (slots)	A	B	C	D	E
800	10	50	100	150	200

### 5.1.2 Caso A2:

O segundo caso proposto também considera o reescalonamento completo da rede, entretanto neste cenário existem dois períodos de publicação diferentes entre os dispositivos. Foram avaliados seis grupos (A-F). O total de dispositivos variou de 10 até 150. A Tabela 9 apresenta os parâmetros deste cenário. O objetivo deste caso é observar o comportamento dos algoritmos com dois períodos diferentes de publicação presentes na rede.

Tabela 9 – Parâmetros do Caso A2.

	Nº de dispositivos					
$P_i$ (slots)	A	B	C	D	E	F
400	5	10	20	30	40	50
800	5	15	30	45	60	100
Total	10	25	50	75	100	150

### 5.1.3 Caso A3:

Neste caso existem quatro períodos de publicação diferentes entre os dispositivos. Foram avaliados seis grupos (A-F) O total de dispositivos variou de 50 a 300. A Tabela 10 apresenta os parâmetros deste cenário.

Tabela 10 – Parâmetros do Caso A3.

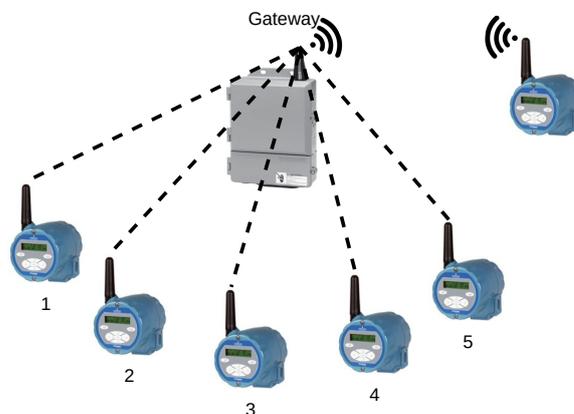
	Nº de dispositivos					
$P_i$ (slots)	A	B	C	D	E	F
200	10	20	20	15	10	5
400	10	20	25	20	20	15
800	10	20	30	25	20	20
1600	20	40	75	140	200	260
Total	50	100	150	200	250	300

## 5.2 Estudo de Caso B: Escalonamento Parcial

O estudo de caso B representa um caso em que um novo dispositivo é agregado a uma rede já formada. Este cenário considera uma rede que já está formada a algum tempo sendo que alguns dispositivos entraram e saíram da rede. Nesta situação, o vetor de escalonamento estará parcialmente preenchido pelos links dos dispositivos que estão agregados a rede. Como alguns dispositivos saíram da rede, existem *timeslots* vagos que podem ser utilizado para a alocação de links de novos dispositivos que desejam agregar a rede. O desafio dos algoritmos é encontrar um escalonamento factível para o novo dispositivo em um vetor de escalonamento que já esta previamente ocupado.

A diferença desta situação para o escalonamento completo da rede (estudo de caso A), é que no caso anterior os dispositivos são escalonados um a um de forma ordenada otimizando a utilização do vetor de escalonamento e o mesmo não acontece neste caso.

Figura 33 – Estudo de Caso B - Escalonamento de um dispositivo agregando-se à rede já formada.



Fonte: do autor.

Para simular o estado do vetor de escalonamento de uma rede que está operando a um determinado tempo e que ocorreu a entrada e saída de diferentes dispositivos, realizou-se o preenchimento prévio do vetor de escalonamento, isto é, foi gerado o escalonamento completo de um conjunto de dispositivos ocupando todo o vetor de escalonamento.

A partir deste vetor gerado, são deletados links do vetor de escalonamento para simular a saída de dispositivos da rede. Após deletar os links, o vetor de escalonamento estará parcialmente preenchido.

Para realizar a exclusão dos links inicialmente é escolhida a porcentagem de ocupação do vetor de escalonamento. A partir da ocupação escolhida, são excluídos links aleatoriamente até atingir a ocupação desejada.

São analisados os casos para diferentes taxas de ocupação, onde a taxa de ocupação é dada pela razão entre o número total de *timeslots* ocupados e o número total de *timeslots* do maior superframe ativo na rede. O vetor de teste é gerado da seguinte forma: é executado o algoritmo LSPA para realizar o escalonamento de um conjunto de dispositivos apresentado na Tabela 11. Na sequência, é definido com qual porcentagem  $y_i \in Y$  de ocupação de banda o teste será realizado sendo que  $Y = [25, 50, 95, 98, 99]$ .

Tabela 11 – Parâmetros do Caso B.

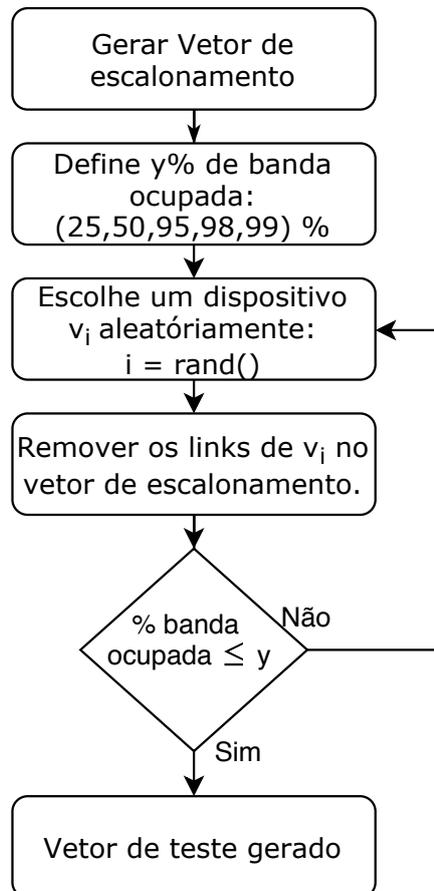
$P_i$ (slots)	Nº Dispositivos
200	30
400	30
800	20

O próximo passo é escolher um dispositivo aleatoriamente do conjunto de dispositivos e apagar os seus links no vetor de escalonamento. Depois de removidos os links, é verificado se a ocupação atual do vetor é menor ou igual a  $y_i$ . Caso a ocupação desejada tenha sido atingida, o vetor de teste foi gerado com sucesso, do contrário, são repetidas as iterações removendo links de dispositivos escolhidos aleatoriamente até se atingir a ocupação desejada. A Figura 34 apresenta a sequência de operações realizadas para gerar o vetor de teste.

Após a criação do vetor de escalonamento, parcialmente preenchido, são executados os algoritmos de escalonamento Han, Han-MO e LSPA para o escalonamento de um dispositivo  $v$  que tem período de  $P_i$ , onde  $P_i$  foi arbitrariamente escolhido como 200 *timeslots*. Este processo é repetido  $n$  vezes e o valor de  $n$  foi escolhido como 100. Como os links apagados do vetor são escolhidos aleatoriamente, cada uma das  $n$  iterações é realizada em um cenário diferente, ou seja, as configurações do vetor escalonamento em cada uma das iterações é diferente.

Neste cenário é analisada a Taxa de Sucesso que representa se os algoritmos con-

Figura 34 – Processo para gerar vetor de escalonamento para testes do estudo de caso B.



Fonte: do autor.

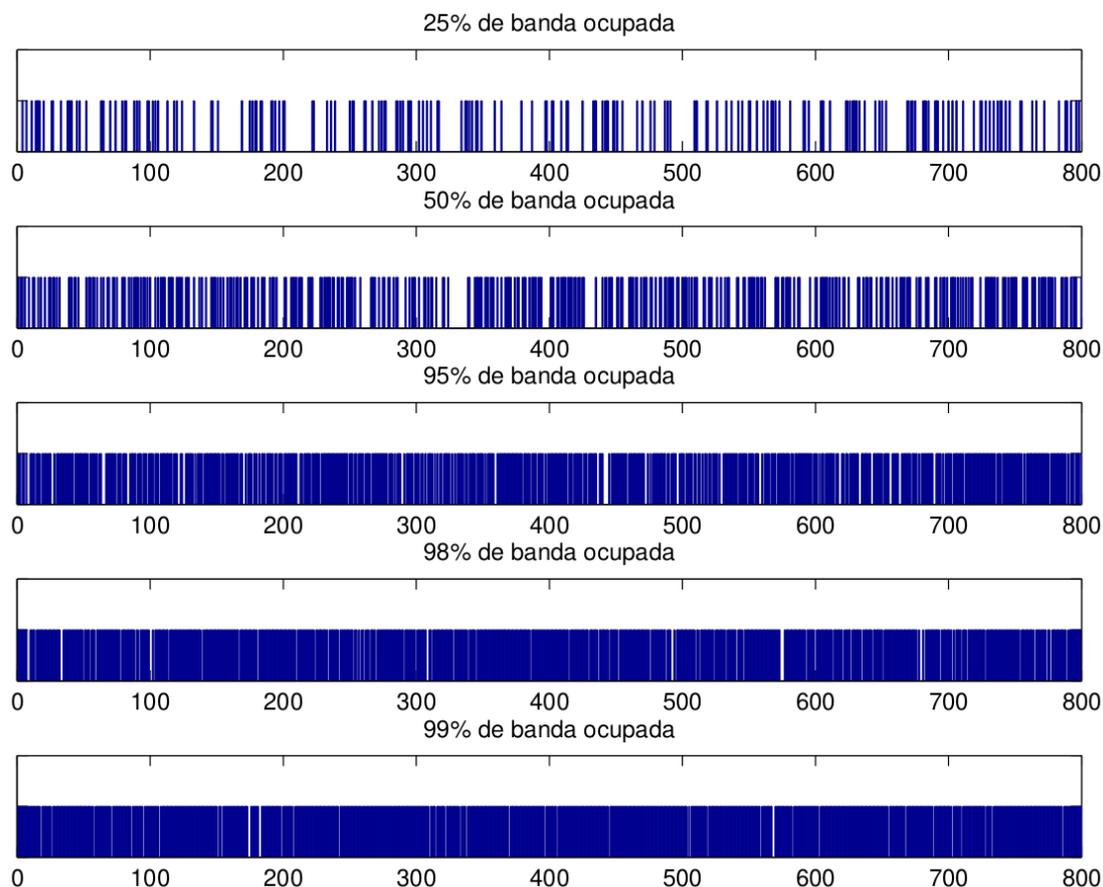
seguem ou não escalonar o dispositivo no vetor de escalonamento parcialmente ocupado. Também é avaliado o tempo de execução de cada algoritmo.

A Figura 35 representa os vetores de escalonamento gerados com as taxas de ocupação de banda. Percebe-se que conforme a taxa de ocupação aumenta, menor é o número de *timeslots* disponíveis. O eixo x da figura representa os *timeslots* do vetor de escalonamento e o eixo y representa se o *timeslot* esta ocupado ou não.

### 5.3 Estudo de Caso C: Múltiplos superframes versus superframe único

Para realizar a comparação entre o processo de escalonamento gerado pelo gerenciador de rede que utiliza múltiplos superframes e o processo de escalonamento gerado pelo gateway comercial que utiliza um único superframe foi realizado um teste

Figura 35 – Exemplo de vetor de escalonamento com diferentes taxas de ocupação.



Fonte: do autor.

prático.

Inicialmente foram configurados três dispositivos com as chaves e o ID da rede do gateway comercial. Estes dispositivos possuem identificações individuais e cada um deles possui um período de publicação diferente. A Tabela 12 apresenta os dispositivos utilizados.

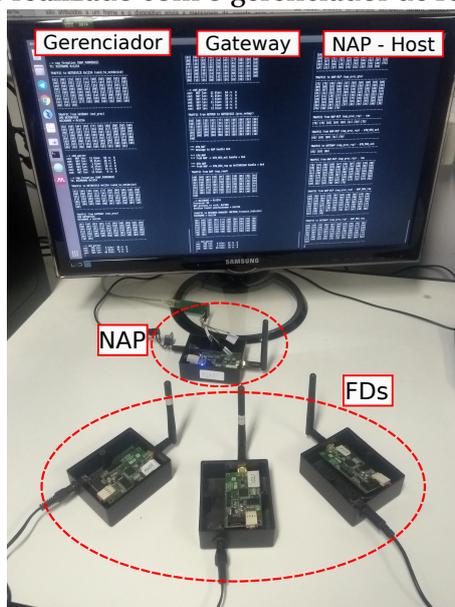
Tabela 12 – Dispositivos utilizados no teste com o gerenciador desenvolvido e com o gerenciador comercial.

ID do dispositivo	Período de publicação (s)
1002	2
1004	4
1008	8

Em seguida, foi iniciado o Gateway Emerson e foram ligados os dispositivos na seguinte sequência: TAG 1008, TAG 1004 e TAG 1002. Os dispositivos foram ligados em

intervalos suficientes para que o processo de agregação de cada um ocorresse sem interferência de outro dispositivo. As mensagens trocadas entre o gateway e os dispositivos durante o processo de agregação foram capturadas pelo sniffer WiAnsys. Os logs do sniffer foram, posteriormente, utilizados nas análises. O mesmo processo foi realizado com o gerenciador proposto. A Figura 36 apresenta a rede formada pelos três dispositivos de campo, o NAP-RCP, o NAP-Host, o gateway e o gerenciador de rede.

Figura 36 – Teste realizado com o gerenciador de rede desenvolvido.



Fonte: do autor.

## 5.4 Métricas

As métricas utilizadas para as análises de desempenho foram:

1. Tempo de execução: é o tempo total necessário para gerar o escalonamento. Foi realizada a medição do tempo necessário para executar a Etapa 4 da Figura 26.
2. Taxa de sucesso: é o número de vezes em que o algoritmo conseguiu escalonar um dispositivo que está entrando na rede sobre o total de experimentos. No momento em que um dispositivo tenta entrar em uma rede já formada o algoritmo de escalonamento obtém sucesso se conseguir encontrar um escalonamento factível.
3. Número de links: é a quantidade de links de dados que o gerenciador envia aos dispositivos no momento do provisionamento.

O capítulo seguinte apresenta os resultados obtidos durante a realização dos estudos de caso.

## 6 RESULTADOS E ANÁLISE

Neste capítulo são apresentados os resultados dos estudos de casos do capítulo anterior. Os testes foram simulados em uma máquina com processador Intel Core 2 Duo de 2,4 GHz e sistema operacional Ubuntu 18.04. Posteriormente são apresentados os links enviados aos dispositivos durante o provisionamento pelo gerenciador desenvolvido que utiliza múltiplos superframes e pelo gerenciador comercial.

### 6.1 Estudo de Caso A:

Nesta seção são apresentados os resultados dos experimentos realizados no estudo de caso A. Como o objetivo é realizar uma comparação entre os algoritmos, sem necessariamente considerar a máquina em que os testes foram realizados, os resultados são apresentados de forma normalizada.

#### 6.1.1 Caso A1

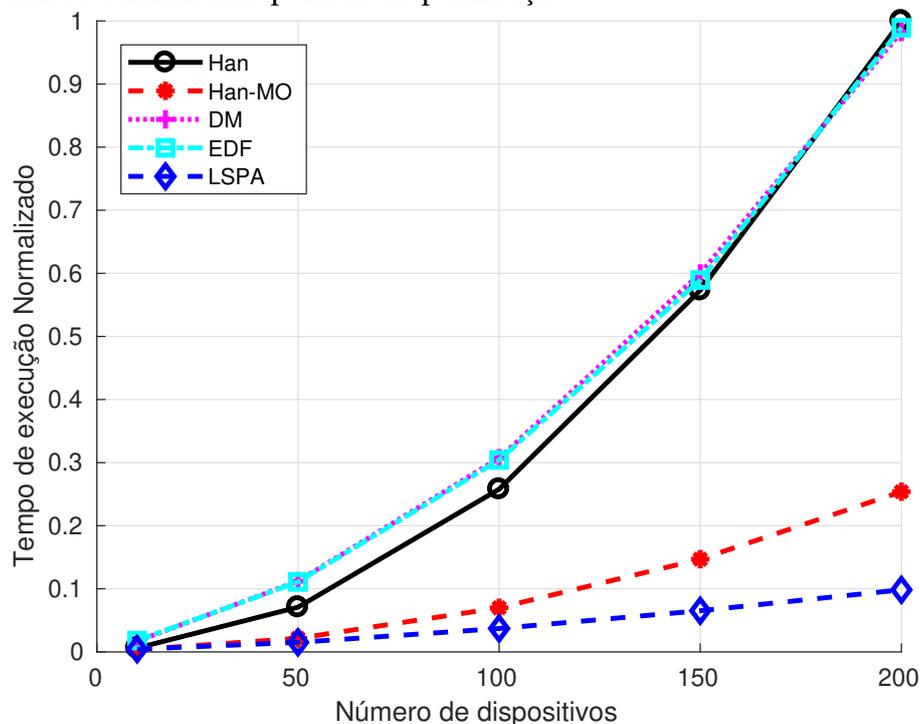
A Figura 37 apresenta os resultados do caso A1. Os gráficos apresentam as médias dos tempos de execução de cada um dos algoritmos utilizando os parâmetros da Tabela 8.

Os algoritmos DM, EDF e Han apresentaram comportamentos semelhantes, sendo os mais lentos entre os algoritmos comparados. Já o método Han-MO, apresentou tempos de execução menores se comparado aos últimos três. O fato de o algoritmo Han-MO alocar todos os links de um dispositivo de uma só vez acelera o processo de escalonamento consideravelmente.

O método LSPA proposto neste trabalho apresentou resultados superiores aos demais métodos. Em comparação com o método de Han-MO (mais rápido entre os comparados), obteve uma redução de até 20% para 50 dispositivos, chegando a 60% de re-

dução para 200 dispositivos mostrando-se uma boa alternativa para os casos onde há apenas um período de publicação na rede.

Figura 37 – Tempo de execução dos algoritmos de escalonamento considerando todos os dispositivos com mesmo período de publicação.



Fonte: do autor.

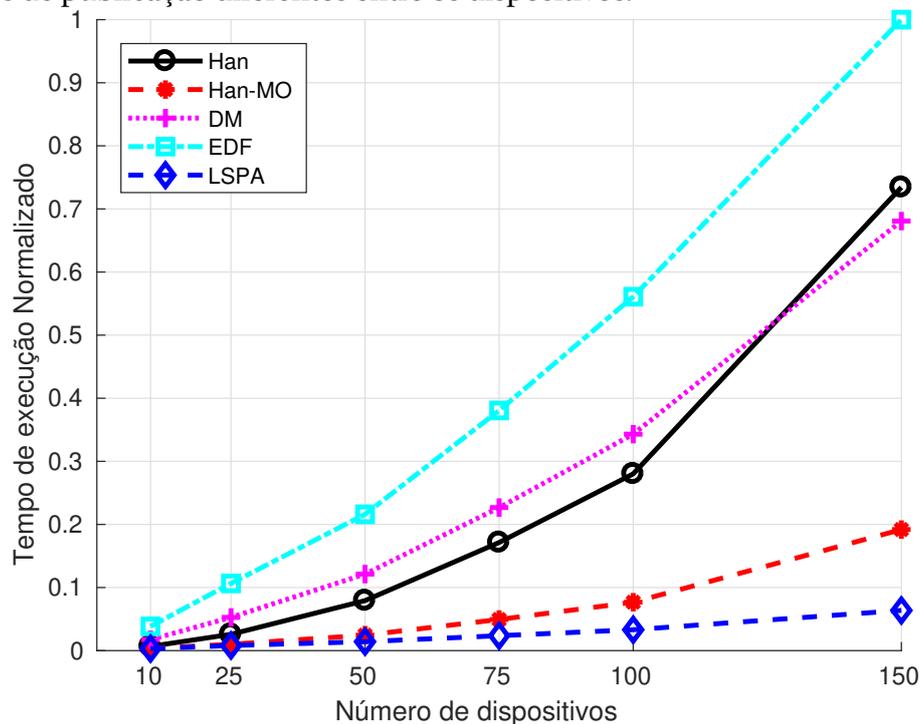
### 6.1.2 Caso A2

A Figura 38 apresenta o resultado do experimento realizado utilizando os parâmetros da Tabela 9.

Neste cenário o EDF apresentou o tempo de execução mais lento dentre os algoritmos e os demais valores foram normalizados a partir dele. Apesar de próximos, o algoritmo de Han apresentou um tempo de execução maior que o DM para 150 dispositivos, o que não ocorreu no caso anterior. O método Han-MO, mais uma vez apresentou tempos de execução menores do que os últimos três.

Neste cenário o método LSPA obteve resultados superiores aos demais algoritmos, sendo que para 150 dispositivos a redução no tempo de execução, em relação ao Han-MO, foi de aproximadamente 65%. A diferença entre eles no caso anterior para 150 dispositivos foi de 50% o que mostra que com dois períodos de publicação na rede a redução obtida pelo LSPA em relação aos demais é maior do que no caso anterior.

Figura 38 – Tempo de execução dos algoritmos de escalonamento considerando dois períodos de publicação diferentes entre os dispositivos.



Fonte: do autor.

### 6.1.3 Caso A3

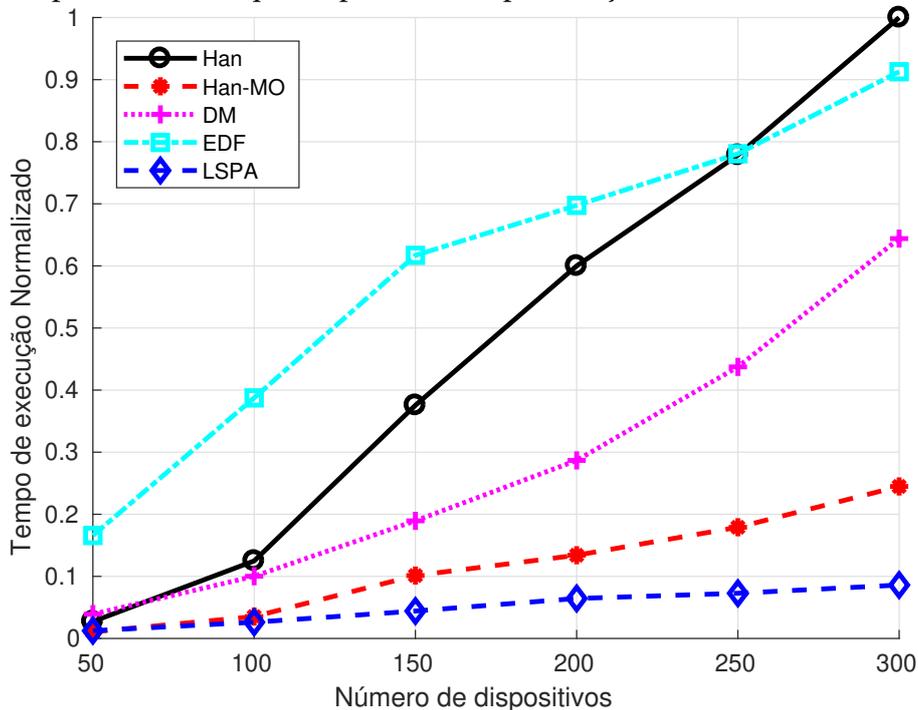
A Figura 39 apresenta o resultado do experimento realizado utilizando os parâmetros da Tabela 10 onde constam 4 períodos diferentes de publicação.

Na maioria dos valores da análise, o EDF apresentou os resultados mais lentos, porém, o algoritmo de Han apresentou resultados próximos ao EDF até se tornar mais lento a partir de 250 dispositivos apresentando o tempo de execução mais lento dentre os algoritmos para 300 dispositivos.

O método Han-MO e o LSPA apresentaram tempos de execução menores do que os demais, repetindo o comportamento dos casos anteriores. Este foi o cenário proposto que mais contém períodos de publicação diferentes e percebe-se que por este motivo os algoritmos variaram seu comportamento em relação aos cenários anteriores. Isso mostra que a quantidade de períodos diferentes presentes na rede influencia o comportamento dos algoritmos.

O LSPA foi, novamente, o mais rápido conseguindo uma redução de 65% em relação ao Han-MO.

Figura 39 – Tempo de execução dos algoritmos de escalonamento considerando o conjunto de dispositivos com quatro períodos de publicação.



Fonte: do autor.

## 6.2 Estudo de Caso B:

A Figura 40 apresenta os resultados do estudo de caso B, onde foram utilizados os parâmetros da Tabela 11.

O gráfico mostra a taxa de sucesso dos algoritmos em diferentes porcentagens de ocupação do vetor de escalonamento. A taxa de sucesso representa o número de cenários em que o algoritmo conseguiu alocar um novo dispositivo sobre o número total de cenários utilizados. Foi variada a ocupação da banda de 25% até 99% e avaliado se o algoritmo de escalonamento conseguiu escalonar o dispositivo que deseja fazer parte da rede ou não.

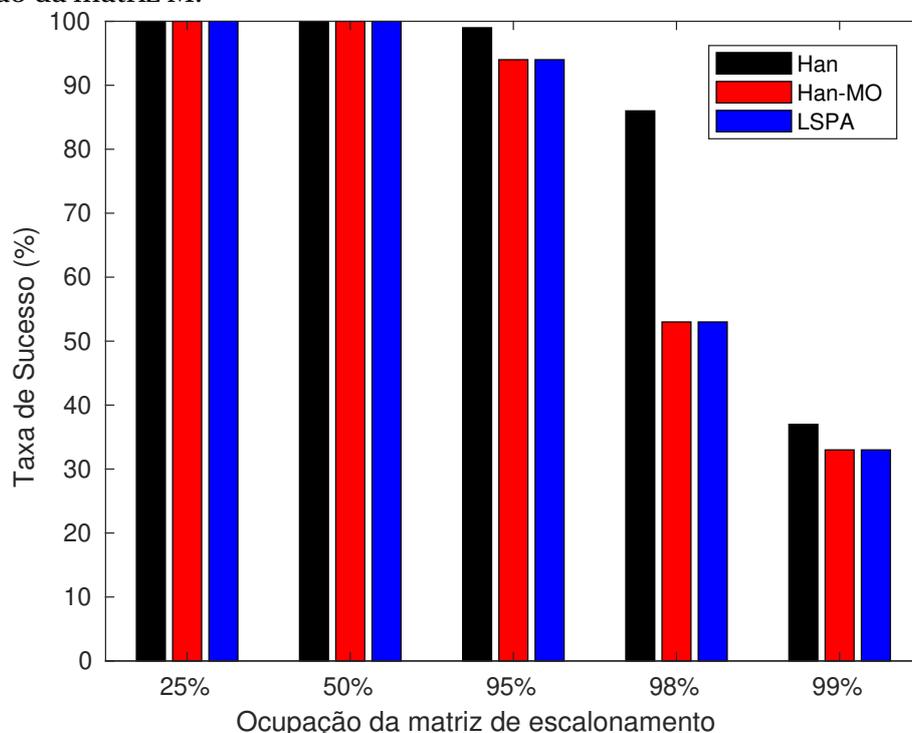
Os resultados mostram que na faixa de ocupação de 25% e 50% do vetor de escalonamento os três algoritmos avaliados conseguiram em todas as repetições escalonar o dispositivo. A partir de 95% os algoritmos Han-MO e o LSPA começam a diminuir a taxa de sucesso. O algoritmo de Han obtém os melhores resultados nesta análise, e isso se dá pelo fato de que a busca por *timeslots* é feita individualmente para cada link tornando o escalonamento mais flexível, ou seja, não existe uma distância fixa entre os links de um dispositivo que está sendo escalonado, o que aumenta a chance do

algoritmo encontrar um escalonamento viável.

Já, os algoritmos Han-MO e o LSPA mantêm uma distância fixa entre os links no momento do escalonamento tornando menores as chances de encontrar um escalonamento factível. Apesar do LSPA ser uma estratégia que torna o processo de escalonamento mais rápido, a taxa de sucesso diminui, pois é menos flexível que o de Han.

Os algoritmos DM e EDF não foram comparados neste cenário, pois consideram prioridade entre os links para realizar o escalonamento, desta forma não são adequados para o escalonamento de um único dispositivo em uma rede já formada.

Figura 40 – Taxa de sucesso dos algoritmos em 100 diferentes cenários para cada % de ocupação da matriz M.

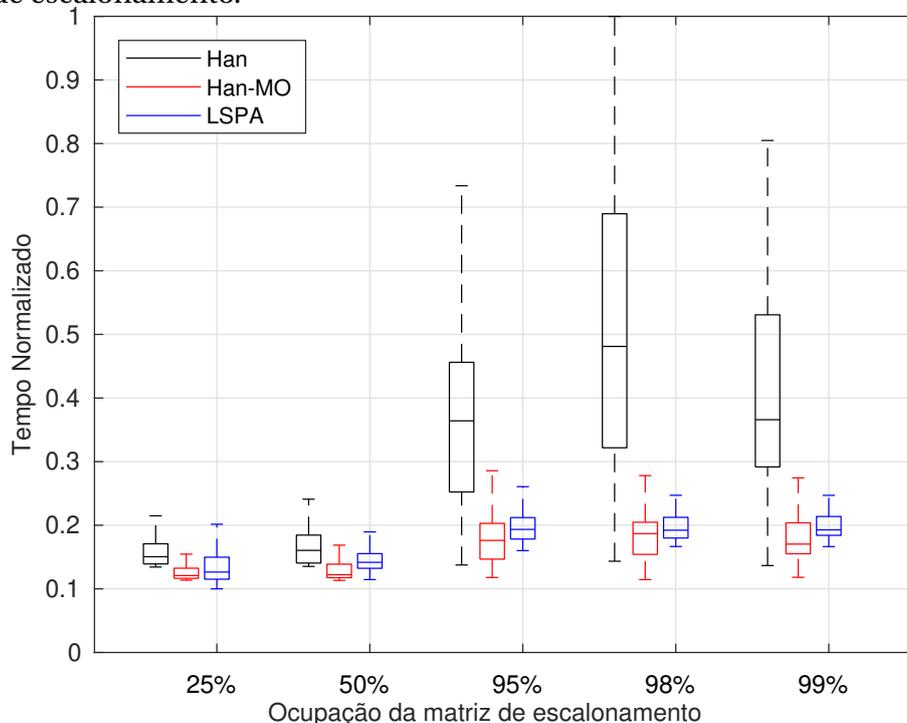


Fonte: do autor.

A Figura 41 apresenta os resultados das execuções dos algoritmos para as diferentes ocupações de banda. Desta vez são apresentadas as distribuições dos resultados, pois cada execução dos algoritmos é realizada considerando uma configuração diferente do vetor de escalonamento. Por exemplo, em 25% de ocupação de banda são realizados testes com 100 diferentes configurações de matriz escalonamento, e da mesma forma para as demais porcentagens. É possível observar que os algoritmos Hahn-MO e o LSPA mantiveram-se com baixa dispersão, tendo seus valores próximos as médias. Já, o algoritmo de Han, apesar de apresentar taxa de sucesso maior, obteve os piores

resultados para os casos em que a matriz de escalonamento esta quase completa, pois teve uma alta dispersão de valores para os casos de 95%, 98% e 99%. Isso se dá pelo fato de que o algoritmo de Han realiza a alocação dos links individualmente e a busca para cada um dos links é realizada em uma área específica do superframe. Desta forma, para as configurações da matriz de escalonamento em que os *timeslots* vagos estão nas primeira posições, o algoritmo precisará de poucas iterações para encontrar um escalonamento para o dispositivo. Porém, caso os *timeslots* vagos estejam distantes das primeiras posições, o algoritmo precisará de diversas iterações para encontrar um escalonamento, uma vez que estas repetições são multiplicadas pelo número de links.

Figura 41 – Tempo de execução para cada algoritmo em diferentes % de ocupação da matriz de escalonamento.



Fonte: do autor.

Os resultados obtidos mostram que conforme o número de dispositivos aumenta, a diferença no tempo de execução entre as técnicas de escalonamento fica cada vez maior. As técnicas clássicas apresentam os piores resultados. Isso se dá pelo fato de a cada iteração do algoritmo a lista de links ser ordenada conforme a prioridade de cada um. Os algoritmos de Han e Han-MO aloca todos os links de um dispositivo sequencialmente. Porém, Han realiza a busca por *timeslots* vagos para cada um dos links individualmente e Han-MO realiza a busca para todos os links simultaneamente, tornando a busca mais rápida. A proposta apresentada não busca *timeslots* disponíveis

como as demais e sim realiza a busca por uma estrutura de *timeslots* disponível. Estas estruturas contém os *timeslots* que serão utilizados pelos links e desta forma a busca é mais rápida que os métodos anteriores.

Em aplicações de automação de fábrica, onde os *timeslots* podem estar na casa de  $\mu s$  o tempo de processamento destas etapas pode representar diversos ciclos dos superframes, causando uma resposta lenta do gerenciador as mudanças na rede. Além disso, considerando uma aplicação real, o gerenciador de rede não possuirá a mesma capacidade de processamento da máquina a qual foram realizados as simulações. Desta forma, é possível que os tempos de execução dos algoritmos comparados sejam ainda maiores. Caso algoritmos de otimização sejam aplicados à RSFI, podem ocorrer sucessivas execuções dos algoritmos de roteamento e escalonamento. A execução sucessiva dos algoritmos de escalonamento pode tornar o gerenciamento um processo lento que eventualmente pode não atender aos requisitos de alta dinamicidade de uma rede aplicada a FA.

### 6.3 Estudo de Caso C:

Este estudo de caso foi realizado em duas partes: inicialmente utilizando-se o gateway comercial e em seguida o mesmo procedimento foi realizado com o gerenciador desenvolvido. Neste estudo foram acompanhando os processos de agregação dos dispositivos utilizando-se o sniffer. Através do sniffer, é possível acompanhar em tempo real a troca de mensagens entre os dispositivos e o ponto de acesso. Após a agregação de cada dispositivo, foi enviado o comando 784 para cada um dos dispositivos para realizar a leitura dos links.

A Tabela 13 apresenta os links de cada dispositivo presentes na rede formada pelo gateway comercial. Observa-se que cada dispositivo recebe uma quantidade diferente de links. Isso se dá pelo fato de que a estratégia de escalonamento do gateway comercial utiliza um superframe de tamanho único com 1024 *timeslots*. Desta forma, para atender aos serviços dos dispositivos o gateway precisa escrever uma quantidade maior de links para aqueles dispositivos que publicam seus dados em um período menor do que 10 segundos.

Para o dispositivo 1002 que publica seus dados a cada 2 segundos, o gateway comercial escreveu 16 links de transmissão normais. Para o dispositivo 1004, escreveu 9

links e para o 1008, escreveu 5 links.

Tabela 13 – Lista de links recebidos pelos dispositivos durante o provisionamento pelo gerenciador comercial.

Dispositivo	Superframe	Timeslot	Ch. Offset	Vizinho	Opção	Tipo
1002	0	20	1	0001	tx	Normal
	0	84	0	0001	tx	Normal
	0	148	1	0001	tx	Normal
	0	212	1	0001	tx	Normal
	0	276	1	0001	tx	Normal
	0	340	0	0001	tx	Normal
	0	404	2	0001	tx	Normal
	0	468	1	0001	tx	Normal
	0	532	0	0001	tx	Normal
	0	596	1	0001	tx	Normal
	0	660	1	0001	tx	Normal
	0	724	1	0001	tx	Normal
	0	788	0	0001	tx	Normal
	0	852	0	0001	tx	Normal
	0	916	1	0001	tx	Normal
	0	980	2	0001	tx	Normal
1004	0	57	0	0001	tx	Normal
	0	121	2	0001	tx	Normal
	0	185	2	0001	tx	Normal
	0	313	1	0001	tx	Normal
	0	441	1	0001	tx	Normal
	0	569	0	0001	tx	Normal
	0	697	1	0001	tx	Normal
	0	825	2	0001	tx	Normal
	0	953	0	0001	tx	Normal
1008	0	33	0	0001	tx	Normal
	0	163	2	0001	tx	Normal
	0	289	2	0001	tx	Normal
	0	545	1	0001	tx	Normal
	0	801	0	0001	tx	Normal

O mesmo teste foi feito para a rede formada pelo gerenciador proposto. Os links de

cada dispositivo são apresentados abaixo na Tabela 14. Percebe-se que para todos os dispositivos foram enviados a mesma quantidade de links (quatro), diferentemente do teste realizado com o gateway comercial.

Tabela 14 – Lista de links recebidos pelos dispositivos durante o provisionamento pelo Gerenciador desenvolvido.

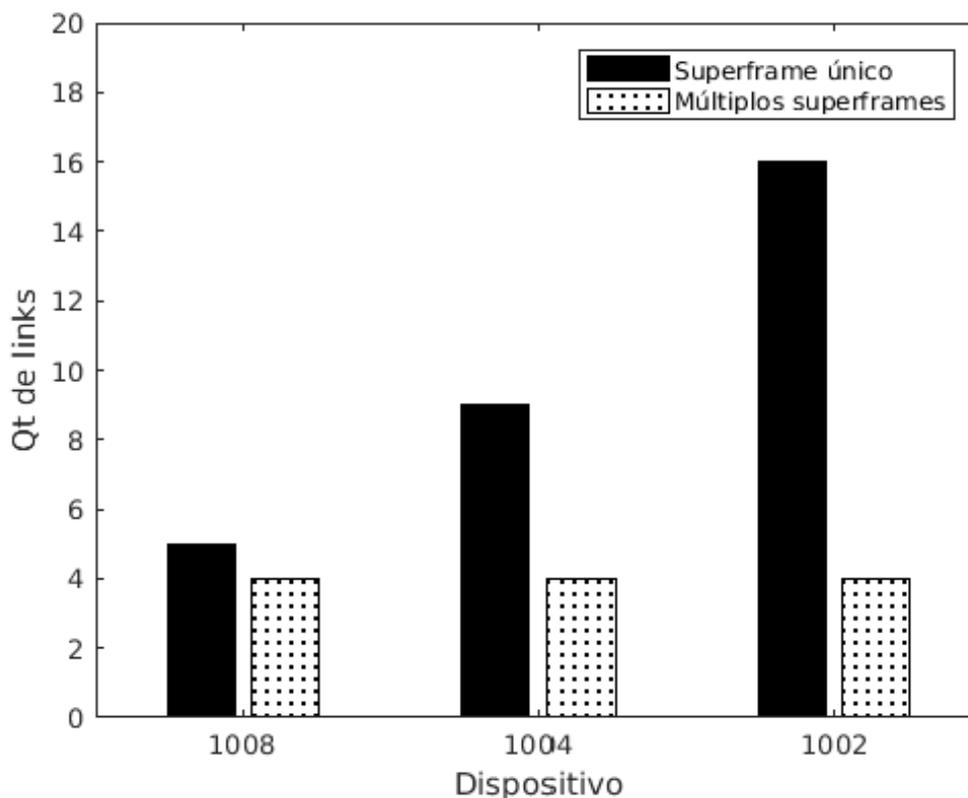
Dispositivo	Superframe	Timeslot	Ch. Offset	Vizinho	Opção	Tipo
1002	4	31	0	0001	tx	Normal
	4	81	0	0001	tx	Normal
	4	131	0	0001	tx	Normal
	4	181	0	0001	tx	Normal
1004	5	32	0	0001	tx	Normal
	5	132	0	0001	tx	Normal
	5	232	0	0001	tx	Normal
	5	332	0	0001	tx	Normal
1008	6	33	0	0001	tx	Normal
	6	233	0	0001	tx	Normal
	6	433	0	0001	tx	Normal
	6	633	0	0001	tx	Normal

Através da Figura 42 é possível visualizar graficamente as diferenças nas quantidades de links que cada dispositivo recebe nos dois métodos comparados. Através desta análise é possível perceber que a estratégia de escalonamento com múltiplos superframe reduz o número de links armazenados por cada dispositivo. As principais vantagens da utilização de múltiplos superframes são a maior eficiência no uso dos recursos (economia de banda), a redução da necessidade de memória, a redução do tráfego na rede durante o provisionamento e manutenção e a economia de energia.

A utilização dos recursos é mais eficiente quando se utiliza múltiplos superframes. A estratégia de utilizar um superframe com tamanho fixo consome mais recursos do que quando utilizam-se múltiplos superframes. Por exemplo, o superframe criado pelo gateway Emerson tem 1024 *timeslots* e quando um dispositivo que publica seus dados a cada 60 segundos é provisionado ele recebe 4 links, ou seja, estes 4 links ocuparão 4 *timeslots* que não serão utilizados por aproximadamente 5 ciclos (51s) do superframe.

A Figura 43 tenta ilustrar a ocupação dos recursos para este caso. A Figura 43(a)

Figura 42 – Quantidades de links recebidos pelos dispositivos utilizando um único superframe e utilizando múltiplos superframes.

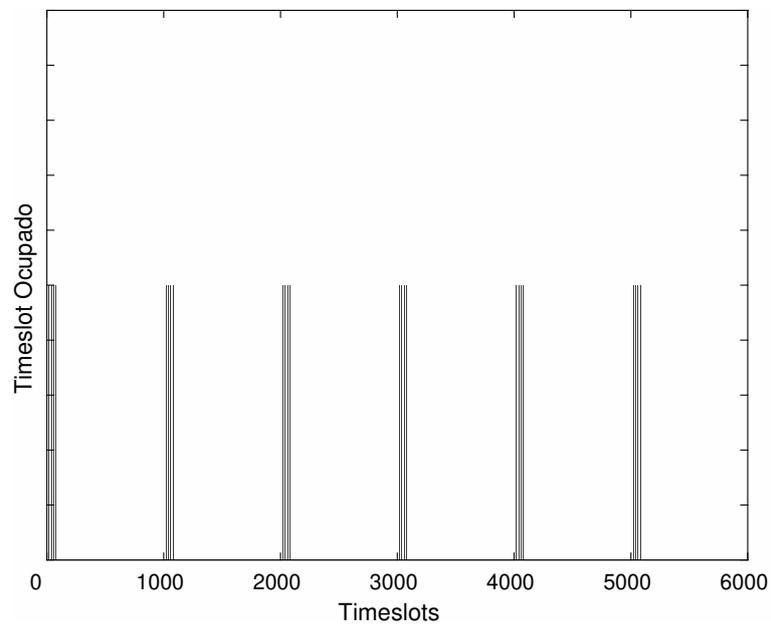


Fonte: do autor.

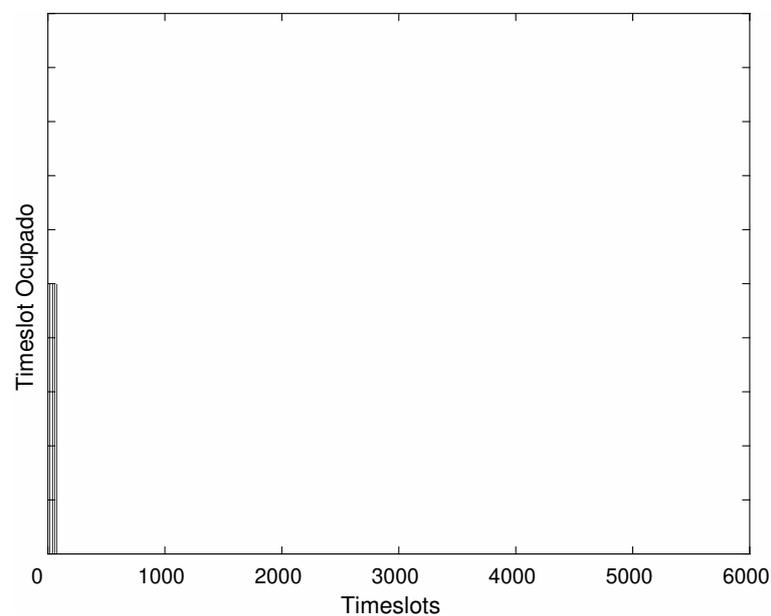
mostra 6 ciclos do superframe de 1024 *timeslots* gerado pelo gateway comercial com os links escritos no dispositivo. Percebe-se que são escritos 4 links no dispositivo dentro do superframe de 1024 *timeslots* e graficamente é possível observar que os links serão repetidos no tempo. Já a Figura 43(b) apresenta um único ciclo de um superframe de 6000 *timeslots*. Ambas as figuras representam um tempo de aproximadamente 60 segundos, e é possível perceber que utilizando-se um superframe específico para os dispositivos que publicam a cada 60 segundos são utilizados menos recursos do que com um único superframe para todos os períodos. Fazendo-se uma razão entre os *timeslots* utilizados sobre o total de *timeslots* chega-se a uma taxa de 0,39% de utilização para o caso de um único superframe e uma taxa de 0,067% para o caso de um superframe específico para o período em questão. Este exemplo considera apenas um dispositivo utilizando os recursos da rede, mas em cenários mais próximos de aplicações reais existirão uma grande quantidade de dispositivos de diferentes períodos de publicação, como apresentado nos cenários anteriores e desta forma a distribuição

eficiente dos recursos é fundamental para o correto funcionamento da rede.

Figura 43 – Ocupação da banda pelos links de comunicação de um dispositivo com período de publicação de 60 segundos.



(a) Superframe único de tamanho fixo para todos os períodos de publicação.



(b) Superframe do tamanho do período específico.

Fonte: do autor.

Com uma quantidade menor de links também diminui a necessidade de memória dos dispositivos para armazenamento de informações. Cada dispositivo que entra na rede recebe informações do gerenciador para que possa operar normalmente. Estas

informações são armazenadas em tabelas as quais os dispositivos consultam quando precisam realizar suas tarefas. Com a utilização de múltiplos superframes os dispositivos podem operar da mesma forma que no caso de um único superframe, porém armazenando uma quantidade menor de informações. Estas informações são enviadas aos dispositivos durante o processo de agregação ou ainda durante etapas de manutenção, desta forma, uma quantidade menor de links reduzirá o tráfego na rede, uma vez que para escrever um link em um dispositivo é necessário enviar um comando, logo, quanto menos links menos comandos enviados aos dispositivos.

Além do uso de *timeslots* de forma ineficiente, existe também a questão de consumo de energia, pois em todos os ciclos do superframe o dispositivo receptor vai acordar para escutar o canal no *timeslot* específico. Utilizando-se o exemplo anterior novamente: em 5 ciclos consecutivos o dispositivo receptor vai acordar e escutar o canal inutilmente uma vez que o transmissor vai precisar de apenas 1 dos ciclos do superframe para realizar a transmissão dos dados.

## 7 CONCLUSÕES

Este trabalho apresentou um estudo sobre algoritmos de escalonamento de links para RSFI. Foi proposto um algoritmo de escalonamento que busca reduzir o tempo necessário para realizar o processo de escalonamento. A técnica proposta realiza um processo de pré-escalonamento dos links. Esta técnica consiste em calcular onde os links serão alocados nos superframes baseando-se na quantidade de links de dados que cada dispositivo recebe e também nos períodos de publicação que a rede suporta. Estas informações são geradas no momento em que o gerenciador de rede começa a operar e são utilizadas todas às vezes em que o gerenciador precisa realizar o processo de escalonamento de um dispositivo.

A proposta foi comparada com outros algoritmos e os resultados mostraram que o tempo de execução dos algoritmos de escalonamento pode ser reduzido quando uma estratégia de pré-escalonamento é utilizada. Este estudo também mostra que o tempo de execução dos algoritmos de escalonamento é uma métrica relevante de ser avaliada, uma vez que dependendo da aplicação da RSFI podem representar um tempo significativo na execução das tarefas de gerenciamento, tornando o gerenciamento da rede pouco eficiente principalmente para aplicações em automação de fábrica onde em alguns protocolos a duração dos *timeslots* pode ser menor que *1 ms*.

Foram realizados três estudos de caso para o reescalonamento completo dos dispositivos da rede. Neste caso os resultados mostraram que o algoritmo proposto é capaz de reduzir o tempo de execução do processo de escalonamento em até 60% nos casos onde existe apenas um período de publicação na rede e em até 65% nos casos em que existem 2 e 4 períodos de publicação na rede.

Um estudo de caso que verifica o comportamento dos algoritmos em um caso onde um novo dispositivo está entrando em uma rede já formada também foi realizado. Fo-

ram analisados casos com diferentes taxas de ocupação de band. O LSPA superou os demais algoritmos na questão do tempo de execução, porém apresentou resultados inferiores ao algoritmo de Han na questão da taxa de sucesso.

Um último estudo de caso realizado foi a utilização de múltiplos superframes frente a utilização de um único superframe de dados. Foi realizada uma comparação entre o escalonamento gerado no protocolo WH por um gateway comercial onde é utilizado um único superframe para alocação de todos os links dos dispositivos e o escalonamento gerado pelo gerenciador de rede desenvolvido. Os resultados mostram que a utilização de múltiplos superframes de dados é uma forma mais eficiente de organizar as comunicações, permitindo uma melhor utilização dos recursos do sistema além de proporcionar economia de energia e de memória para os dispositivos.

Resumindo, as principais contribuições desta dissertação são a elaboração de uma técnica de escalonamento específica para RFSI, a comparação entre diferentes estratégias de escalonamento, avaliando o tempo de execução e a comparação entre múltiplos superframes e superframe único de dados. Ainda, como legado deste trabalho, foi concluído o desenvolvimento de um gateway, ponto de acesso e gerenciador de rede compatível com o protocolo WH, que permitiu avaliar na prática as estratégias de escalonamento e que permitirá a continuidade de estudos e desenvolvimentos com este protocolo.

Como possíveis trabalhos futuros, de continuidade deste, sugerem-se o aperfeiçoamento da técnica de pré-escalonamento visando atender também topologias mais complexas como árvore e malha. A adaptação do pré-escalonamento nestes tipos de topologias pode abrir espaço para o reuso de frequência, onde duas ou mais transmissões poderiam ocorrer em um mesmo *timeslot* em canais diferentes.

Outrossim, a implementação das demais tarefas de gerenciamento, como por exemplo o roteamento, em um simulador ou em uma rede real pode contribuir para o entendimento dos efeitos que o tempo de execução dos algoritmos de escalonamento causam na tomada de decisão por parte do gerenciador.

## REFERÊNCIAS

5G-ACIA. ALLIANCE FOR CONNECTED INDUSTRIES AND AUTOMATION. **5G for connected industries and automation**. [S.l.]: ZVEI – German Electrical and Electronic Manufacturers' Association, 2019a. Disponível em: <https://www.5g-acia.org/publications/5g-for-connected-industries-and-automation-white-paper/> Acesso em: 15 Jul 2019.

5G-ACIA. ALLIANCE FOR CONNECTED INDUSTRIES AND AUTOMATION. **5G for automation in industry: primary use cases, functions and service requirements**. [S.l.]: ZVEI – German Electrical and Electronic Manufacturers' Association, 2019b. Disponível em: <https://www.5g-acia.org/publications/5g-for-automation-in-industry-white-paper/> Acesso em: 15 Jul 2019.

AGIWAL, M.; ROY, A.; SAXENA, N. Next generation 5G wireless networks: a comprehensive survey. **IEEE Communications Surveys and Tutorials**, [S.l.], v. 18, n. 3, p. 1617–1655, Julho 2016.

AL-NIDAWI, Y.; YAHYA, H.; KEMP, A. Tackling mobility in low latency deterministic multihop IEEE 802.15.4e sensor network. **IEEE Sensors Journal**, [S.l.], v. 16, p. 1412–1427, Março 2016.

BHATIA, A.; HANSDAH, R. C. A Distributed TDMA slot scheduling algorithm for spatially correlated contention in WSNs. *In*: INTERNATIONAL CONFERENCE ON ADVANCED INFORMATION NETWORKING AND APPLICATIONS WORKSHOPS, 27., 2013, Barcelona. **Proceedings [...]** IEEE, 2013. p. 377–384.

DEZFOULI, B.; RADI, M.; CHIPARA, O. Mobility-aware real-time scheduling for low-power wireless networks. *In*: IEEE INTERNATIONAL CONFERENCE ON COMPUTER COMMUNICATIONS (INFOCOM), 35., 2016, São Francisco-CA, (EUA). **Proceedings [...]** IEEE, 2016. p. 1–9.

DEZFOULI, B.; RADI, M.; CHIPARA, O. REWIMO: a real-time and reliable low-power wireless mobile network. **ACM Transactions on Sensor Networks (TOSN)**, [S.l.], v. 13, p. 17:1–17:47, Agosto 2017.

DICKOW, V. H. **Avaliação de algoritmos de roteamento e escalonamento de mensagens para redes wireless**. 2014. 88 p. Dissertação (Mestrado em Engenharia Elétrica) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2014.

DJUKIC, P.; VALAEE, S. Delay aware link scheduling for multi-hop TDMA wireless networks. **IEEE/ACM Transactions on Networking**, [S.l.], v. 17, n. 3, p. 870–883, Junho 2009.

EMERSON, E. C. **Process Management, Gateway Smart Wireless**. [S.l.: s.n.], 2013. Disponível em: <https://www.emerson.com/> Acesso em: 10 Jun 2019.

ERGEN, S. C.; VARAIYA, P. TDMA scheduling algorithms for wireless sensor networks. **Wireless Networks**, [S.l.], v. 16, n. 4, p. 985–997, Maio 2010.

FOUNDATION, H. C. **HCF\_SPEC-290, Revision 1.1 WirelessHART Device Specification**. Austin-TX,(EUA): [s.n.], 2008.

GANGAKHEDKAR, S. *et al.* Use cases, requirements and challenges of 5G communication for industrial automation. *In: IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS WORKSHOPS (ICC WORKSHOPS)*, 2018, Kansas City-MO,(EUA). **Proceedings [...]** IEEE, 2018. p. 1–6.

HAHN, D. H. **Desenvolvimento de um ponto de acesso para redes WirelessHart**. 2011. 57 p. Projeto de diplomação (Graduação em Engenharia Elétrica) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2011.

HAN, S. *et al.* Reliable and real-time communication in industrial wireless mesh networks. *In: IEEE REAL-TIME AND EMBEDDED TECHNOLOGY AND APPLICATIONS SYMPOSIUM (RTAS)*, 17., 2011, Massachusetts (EUA). **Proceedings [...]** IEEE Computer Society, 2011. p. 3–12.

KRÖTZ, C. A. **Ferramenta e método para obtenção de parâmetros de confiabilidade fim-a-fim de redes industriais sem fio**. 2019. 71 p. Dissertação (Mestrado em Engenharia Elétrica) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2019.

KÜNZEL, G.; INDRUSIAK, L.; PEREIRA, C. E. Latency and lifetime enhancements in IWSN: a q-learning approach for graph routing. **IEEE Transactions on Industrial Informatics**, [S.l.], 2019.

LIANG, W. *et al.* WIA-FA and its applications to digital factory: a wireless network solution for factory automation. **Proceedings of the IEEE**, [S.l.], v. 107, n. 6, p. 1–21, Fevereiro 2019.

LUVISOTTO, M.; PANG, Z.; DZUNG, D. Ultra high performance wireless control for critical applications: challenges and directions. **IEEE Transactions on Industrial Informatics**, [S.l.], v. 13, n. 3, p. 1448–1459, Junho 2017.

LUVISOTTO, M.; PANG, Z.; DZUNG, D. High-Performance wireless networks for industrial control applications: new targets and feasibility. **Proceedings of the IEEE**, [S.l.], v. 107, n. 6, p. 1074–1093, Março 2019.

LUVISOTTO, M. *et al.* Physical layer design of high-performance wireless transmission for critical control applications. **IEEE Transactions on Industrial Informatics**, [S.l.], v. 13, n. 6, p. 2844–2854, Dezembro 2017.

MIN, M. *et al.* Traffic aware multiple slotframes scheduling algorithm in industrial IoT applications using IEEE802.15.4e TSCH. **International Conference on Communication Technology Proceedings, ICCT**, [S.l.], p. 608–614, Outubro 2016.

MODEKURTHY, V. P.; SAIFULLAH, A.; MADRIA, S. DistributedHART: a distributed real-time scheduling system for wirelessHART networks. *In: IEEE REAL-TIME AND EMBEDDED TECHNOLOGY AND APPLICATIONS SYMPOSIUM (RTAS)*, 25., 2019, Montreal (Canadá). **Proceedings [...]** IEEE, 2019. p. 216–227.

MÜLLER, I. **Gerenciamento descentralizado de redes sem fio industriais segundo o padrão WirelessHART**. 2012. 104 p. Tese (Doutorado em Engenharia Elétrica) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2012.

MULLER, I. *et al.* Development of a WirelessHART compatible field device. *In: IEEE INTERNATIONAL INSTRUMENTATION AND MEASUREMENT TECHNOLOGY CONFERENCE (I2MTC)*, 2010. **Proceedings [...]** [S.l.: s.n.], 2010. p. 1430–1434.

MÜLLER, I. *et al.* WirelessHART Fast Collect: a decentralized approach for intermittent field devices. *In: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL INFORMATICS (INDIN)*, 11., 2013. **Proceedings [...]** [S.l.: s.n.], 2013. p. 254–259.

NETTO, J. C. **Scheduling by Macro-Operators and some Applications to Robot Control**. 1995. 100 p. Tese (Doutorado em Engenharia Elétrica) — Université Catholique de Louvain, Louvain, Bélgica, 1995.

NOBRE, M.; SILVA, I.; GUEDES, L. A. Routing and scheduling algorithms for wirelessHART networks: a survey. **Sensors (Switzerland)**, [S.l.], v. 15, n. 5, p. 9703–9740, Maio 2015.

PANG, Z.; LUVISOTTO, M.; DZUNG, D. Wireless high-performance communications: the challenges and opportunities of a new target. **IEEE Industrial Electronics Magazine**, [S.l.], v. 11, n. 3, p. 20–25, Setembro 2017.

POURIA, Z. *et al.* Implementation of wirelessHART in the NS-2 simulator and validation of its correctness. **Sensors (Switzerland)**, [S.l.], v. 14, n. 5, p. 8633–8668, Maio 2014.

R. TELES HERMETO, A. G.; THEOLEYRE, F. Scheduling for IEEE802.15.4-TSCH and slow channel hopping MAC in low power industrial wireless networks: a survey. **Computer Communications**, [S.l.], v. 114, p. 84–105, Novembro 2017.

RECH, J. R. **Desenvolvimento de um gerente de rede WirelessHART**. 2012. 55 p. Projeto de diplomação (Graduação em Engenharia Elétrica) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2012.

SAIFULLAH, A. *et al.* Real-time scheduling for WirelessHART networks. *In: IEEE REAL-TIME SYSTEMS SYMPOSIUM*, 31., 2010. **Proceedings [...]** [S.l.: s.n.], 2010. p. 150–159.

SCHEIBLE, G. *et al.* Unplugged but connected - design and implementation of a truly wireless real-time sensor/actuator interface. **IEEE Industrial Electronics Magazine**, [S.l.], v. 1, n. 2, p. 25–34, Janeiro 2007.

VALADÃO, Y. *et al.* Industrial Wireless Automation: overview and evolution of wia-pa. **IFAC-PapersOnLine**, [S.l.], v. 51, n. 10, p. 175–180, Agosto 2018.

ZAND, P. *et al.* Wireless industrial monitoring and control networks: the journey so far and the road ahead. **Journal of Sensor and Actuator Networks**, [S.l.], v. 1, n. 2, p. 123–152, Agosto 2012.

ZHU, H. *et al.* IETF IoT based wireless communication for latency-sensitive use cases in building automation. *In*: IEEE INTERNATIONAL SYMPOSIUM ON INDUSTRIAL ELECTRONICS (ISIE), 25., 2016, Santa Clara-CA (EUA). **Proceedings [...]** IEEE, 2016. p. 1168–1173.