



**GILSON MIRANDA JÚNIOR**

**GEN3: PROTOCOLO MAC ADAPTATIVO ESCALONADO  
COM MULTIPLEXAÇÃO ESTATÍSTICA**

**LAVRAS – MG**

**2017**

**GILSON MIRANDA JÚNIOR**

**GEN3: PROTOCOLO MAC ADAPTATIVO ESCALONADO COM  
MULTIPLEXAÇÃO ESTATÍSTICA**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, área de concentração em Redes de Computadores e Sistemas Embarcados, para a obtenção do título de Mestre.

Prof. PhD Luiz Henrique Andrade Correia

Orientador

**LAVRAS – MG**

**2017**

**Ficha catalográfica elaborada pelo Sistema de Geração de Ficha Catalográfica da Biblioteca  
Universitária da UFLA, com dados informados pelo(a) próprio(a) autor(a)**

Júnior, Gilson Miranda

Gen3: Protocolo MAC Adaptativo Escalonado com Multiplexação Estatística / Gilson Miranda Júnior. 1 – Lavras : UFLA, 2017.

132 p. : il.

Dissertação (mestrado acadêmico)–Universidade Federal de Lavras, 2017.

Orientador: Prof. PhD Luiz Henrique Andrade Correia.

Bibliografia.

1. Protocolo MAC. 2. TDMA. 3. Multiplexação Estatística.  
I. Correia, Luiz Henrique Andrade.. II. Título.

**GILSON MIRANDA JÚNIOR**

**GEN3: PROTOCOLO MAC ADAPTATIVO ESCALONADO COM  
MULTIPLEXAÇÃO ESTATÍSTICA**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, área de concentração em Redes de Computadores e Sistemas Embarcados, para a obtenção do título de Mestre.

APROVADA em 13 de Abril de 2017.

José Marcos Silva Nogueira UFMG  
Daniel Fernandes Macedo UFMG



Prof. PhD Luiz Henrique Andrade Correia  
Orientador

**LAVRAS – MG  
2017**

## RESUMO

Redes de Sensores Sem Fio (RSSF) têm sido objeto de estudo e desenvolvimento há mais de uma década. Com o desenvolvimento de dispositivos mais robustos e com mais recursos, as propostas para uso de RSSF passaram de aplicações simples, com alta tolerância a falhas, para aplicações de monitoramento e controle que requerem alta disponibilidade e confiabilidade. Os protocolos de controle de acesso ao meio (MAC - *Medium Access Control*) têm papel determinante para que essas redes operem de forma correta e eficiente.

Os protocolos MAC podem ser classificados como baseados em contenção (CBP - *Contention-Based Protocol*) ou escalonados (TDMA - *Time Division Multiple Access*). Protocolos CBP apresentam baixa latência na entrega dos pacotes em relação aos protocolos TDMA em redes com baixo tráfego. Tráfego intenso ou alta densidade de nós, no entanto, prejudicam a operação de protocolos CBP, fazendo com que tenham operação imprevisível, com maiores perdas e latência mais alta que protocolos TDMA.

Apesar de apresentarem comportamento mais estável que protocolos CBP quando submetidos a tráfego intenso, protocolos TDMA podem subutilizar o meio de transmissão em redes com baixo tráfego ou quando os nós apresentam geração de tráfego desbalanceada. Se em uma rede TDMA um nó gera poucos pacotes e outro gera muitos pacotes, o tempo de uso do meio se mantém igual para ambos, sendo subutilizado pelo primeiro nó e potencialmente insuficiente para o segundo nó. A utilização de multiplexação estatística em protocolos TDMA pode atenuar esse problema, por permitir o uso oportunístico de *slots* subutilizados.

Outra questão a ser considerada é a mobilidade e adaptação da rede a diferentes condições. Em geral, protocolos TDMA mantêm informações sobre a rede que variam com a topologia e a movimentação de nós. Alguns protocolos definem parâmetros com base em situações esperadas da rede que podem limitar a operação dos nós.

Este trabalho apresenta o Gen3, um protocolo TDMA generalista que organiza a rede logicamente como uma árvore. O protocolo é autoconfigurável e se adapta à quantidade de nós da rede enquanto mantém a operação dos nós já associados. O protocolo emprega múltiplos canais para aumentar a quantidade de comunicações simultâneas, visando o incremento da vazão, da taxa de entrega e a redução da latência de entrega dos pacotes. Além disso, um mecanismo de multiplexação estatística permite que o tempo ocioso de um *slot* pertencente a um nó seja aproveitado por outro nó que tenha demanda pelo meio de transmissão.

O Gen3 foi avaliado por meio de exaustivas simulações utilizando o simulador OM-NeT++ e o *framework* INET. Os resultados obtidos mostram sua capacidade de oferecer altas taxas de entrega e vazão, mesmo quando submetido a tráfego intenso, superando o protocolo utilizado para comparação. A adaptabilidade do protocolo pôde ser avaliada pela sua correta configuração em diferentes topologias e pelo impacto reduzido da mobilidade dos nós nas métricas de transmissão.

**Palavras-chave:** Protocolo MAC; TDMA; Multiplexação Estatística

## ABSTRACT

Wireless Sensor Networks (WSNs) have been subject of development and study for more than a decade. With the development of more robust devices that integrate more resources, proposals for using RSSF have passed from applications with high fault tolerance to control and monitoring applications that require high availability and reliability. Medium Access Control (MAC) protocols play a key role in these networks, enabling them to operate reliably and efficiently.

MAC protocols can be classified as Contention-Based Protocols (CBP) or scheduled protocols (TDMA - Time Division Multiple Access). CBP protocols present low delivery latency compared to TDMA protocols in low traffic conditions. Under high traffic or high node densities, however, CPB protocols present unpredictable operation, with higher packet losses and increased latency compared to TDMA protocols.

Although they present a more stable behavior over CBP protocols when submitted to heavy traffic, TDMA protocols may underutilize the medium in low traffic conditions or when nodes present unbalanced traffic generation. If in a TDMA network a node generates few packets and another one generates a high amount of packets, the time allocated to each node remains equal, being underutilized by the first node and potentially insufficient for the second. The use of statistical multiplexing in TDMA protocols can attenuate this problem by allowing the opportunistic usage of underutilized slots.

Another issue to consider is the mobility and network adaptation to different conditions. In general, TDMA protocols maintain information about the network that vary with topology and node movement. Some protocols define parameters based on expected network characteristics, which may eventually limit nodes operation or even their association to the network.

This work presents Gen3, a generalist TDMA MAC protocol that organizes the network logically as a tree. The protocol is autoconfigurable and adaptive to the number of nodes in the network while maintaining the operation of already associated nodes. By using multiple channels, it allows collision-free concurrent communications to increase throughput, delivery rate, and reduce packet latency. In addition, a statistical multiplexing mechanism allows the opportunistic use of an underutilized slot by nodes that have higher demand for medium access.

Gen3 was evaluated through exhaustive simulations using OMNeT++ simulator and INET framework. Results show its capacity to offer high delivery rates and high throughput, even when submitted to intense traffic, achieving higher performance than the protocol used for comparison. Protocol's adaptability could be evaluated by its correct configuration in different topologies and by the reduced impact of node mobility in packet transmission metrics.

**Keywords:** MAC Protocol; TDMA; Statistical Multiplexing

## LISTA DE FIGURAS

Figura 2.1 – Tipos de topologia . . . . .	19
Figura 2.2 – Organização em camadas com serviços <i>cross-layer</i> . . . . .	20
Figura 2.3 – Tipos de divisão de tempo em protocolos TDMA. . . . .	22
Figura 2.4 – Marcação de tempo em nós sensores. . . . .	23
Figura 2.5 – Multiplexação estatística em rede de telefonia e dados. . . . .	29
Figura 3.1 – Formato de divisão de tempo do Gen3. . . . .	41
Figura 3.2 – Alocação de <i>frames</i> de um nó pai para seus nós filhos. . . . .	43
Figura 3.3 – Alocação de <i>frames</i> e <i>slots</i> para comunicação livre de colisões. . . . .	47
Figura 3.4 – Alocação alternada de <i>slots</i> e canais. . . . .	50
Figura 3.5 – Formato do pacote de controle. . . . .	51
Figura 3.6 – Formato de endereçamento utilizado no trabalho. . . . .	53
Figura 3.7 – Troca de mensagens para a associação de um novo nó. . . . .	55
Figura 3.8 – Condições testadas para determinar a desassociação. . . . .	58
Figura 3.9 – Condições para envio de pacote de desconexão. . . . .	59
Figura 3.10 – Comunicação entre nó filho (B) e nó pai (A) durante um <i>slot</i> . . . . .	61
Figura 3.11 – Formato do pacote de dados. . . . .	62
Figura 3.12 – Transmissões utilizando MXC. . . . .	64
Figura 4.1 – Topologias regulares com 25 nós e nó 12 como <i>sink</i> . . . . .	70
Figura 4.2 – Topologias lineares com 12 e 23 nós. . . . .	71
Figura 4.3 – Topologia 9F1 - 9 nós fixos e 1 nó com mobilidade controlada. . . . .	71
Figura 4.4 – Formato de descrição de cenários. . . . .	74
Figura 4.5 – Cenários de avaliação de associação e estabilização. . . . .	74
Figura 4.6 – Cenários de avaliação de transmissão de dados. . . . .	75
Figura 4.7 – Cenários de avaliação de mobilidade. . . . .	75
Figura 4.8 – Cenários de avaliação de consumo energético. . . . .	75
Figura 4.9 – Divisão de tempo adotada pelo MC-LMAC. . . . .	77
Figura 4.10 – Formato dos <i>slots</i> do MC-LMAC. . . . .	78
Figura 4.11 – Diagrama de estados do MC-LMAC. . . . .	79
Figura 4.12 – Pacote de controle do MC-LMAC. . . . .	80
Figura 5.1 – Cenário de configuração avaliado: associação. . . . .	84
Figura 5.2 – Tempo médio de associação na topologia 25D25. . . . .	85

Figura 5.3 – Tempo médio de associação na topologia 25D75. . . . .	86
Figura 5.4 – Tempo médio de associação na topologia 25D100. . . . .	87
Figura 5.5 – Tempo médio de associação na topologia 25D150. . . . .	87
Figura 5.6 – Tempo médio de associação na topologia L12. . . . .	88
Figura 5.7 – Tempo médio de associação na topologia L23. . . . .	88
Figura 5.8 – Cenário de configuração avaliado: estabilização. . . . .	89
Figura 5.9 – Tempo médio de estabilização na topologia 25D25. . . . .	90
Figura 5.10 – Tempo médio de estabilização na topologia 25D75. . . . .	90
Figura 5.11 – Tempo médio de estabilização na topologia 25D100. . . . .	91
Figura 5.12 – Tempo médio de estabilização na topologia 25D150. . . . .	91
Figura 5.13 – Tempo médio de estabilização na topologia L12. . . . .	92
Figura 5.14 – Tempo médio de estabilização na topologia L23. . . . .	92
Figura 5.15 – Cenário de transmissão avaliado: taxa de entrega com tráfego CBR. . . . .	95
Figura 5.16 – Entrega de pacotes na topologia 25D25 com tráfego CBR. . . . .	96
Figura 5.17 – Entrega de pacotes na topologia 25D150 com tráfego CBR. . . . .	97
Figura 5.18 – Entrega de pacotes na topologia L12 com tráfego CBR. . . . .	97
Figura 5.19 – Entrega de pacotes na topologia L23 com tráfego CBR. . . . .	98
Figura 5.20 – Cenário de transmissão avaliado: vazão com tráfego CBR. . . . .	99
Figura 5.21 – Vazão média de pacotes na topologia 25D25 com tráfego CBR. . . . .	99
Figura 5.22 – Vazão média de pacotes na topologia 25D150 com tráfego CBR. . . . .	100
Figura 5.23 – Vazão média de pacotes na topologia L12 com tráfego CBR. . . . .	101
Figura 5.24 – Vazão média de pacotes na topologia L23 com tráfego CBR. . . . .	101
Figura 5.25 – Cenário de transmissão avaliado: latência com tráfego CBR. . . . .	102
Figura 5.26 – Distribuição de latências na topologia 25D25 com tráfego CBR. . . . .	103
Figura 5.27 – Distribuição de latências na topologia 25D150 com tráfego CBR. . . . .	103
Figura 5.28 – Distribuição de latências na topologia L12 com tráfego CBR. . . . .	104
Figura 5.29 – Distribuição de latências na topologia L23 com tráfego CBR. . . . .	104
Figura 5.30 – Cenário de transmissão avaliado: taxa de entrega com tráfego VBR. . . . .	105
Figura 5.31 – Entrega de pacotes com tráfego VBR. . . . .	106
Figura 5.32 – Cenário de transmissão avaliado: vazão com tráfego VBR. . . . .	106
Figura 5.33 – Vazão média de pacotes com tráfego VBR. . . . .	107
Figura 5.34 – Cenário de transmissão avaliado: latência com tráfego VBR. . . . .	107



Figura 5.35 – Distribuição de latências na topologia 25D25 com tráfego VBR. . . . .	108
Figura 5.36 – Distribuição de latências na topologia 25D150 com tráfego VBR. . . . .	109
Figura 5.37 – Distribuição de latências na topologia L12 com tráfego VBR. . . . .	109
Figura 5.38 – Distribuição de latências na topologia L23 com tráfego VBR. . . . .	110
Figura 5.39 – Cenário de mobilidade avaliado: topologia 9F1. . . . .	111
Figura 5.40 – Taxa de entrega na topologia 9F1. . . . .	112
Figura 5.41 – Vazão média no <i>sink</i> com a topologia 9F1. . . . .	113
Figura 5.42 – Distribuição de latências na topologia 9F1. . . . .	113
Figura 5.43 – Tempo médio de <i>handoff</i> na topologia 9F1. . . . .	114
Figura 5.44 – Cenário de mobilidade avaliado: topologia 9F11. . . . .	115
Figura 5.45 – Taxa de entrega na topologia móvel 9F11. . . . .	116
Figura 5.46 – Distribuição de latências na topologia 9F11. . . . .	117
Figura 5.47 – Cenário avaliado: consumo energético e MXC. . . . .	119
Figura 5.48 – Consumo médio de energia na topologia 25D25. . . . .	121
Figura 5.49 – Consumo médio de energia na topologia 25D150. . . . .	121
Figura 5.50 – Consumo médio de energia na topologia L12. . . . .	122
Figura 5.51 – Consumo médio de energia na topologia L23. . . . .	122
Figura 5.52 – Latência com diferentes limiares de MXC na topologia 25D25. . . . .	123
Figura 5.53 – Latência com diferentes limiares de MXC na topologia 25D150. . . . .	124

## LISTA DE TABELAS

Tabela 2.1 – Sumarização dos trabalhos analisados. . . . .	38
Tabela 3.1 – Notação utilizada para a descrição do protocolo Gen3. . . . .	41
Tabela 3.2 – Comandos do pacote de controle. . . . .	51
Tabela 4.1 – Parâmetros de transmissão e do protocolo Gen3. . . . .	69
Tabela 4.2 – Sumário das topologias utilizadas nos testes. . . . .	72
Tabela 5.1 – Parâmetros complementares do Gen3 para os testes de configuração. . . . .	83
Tabela 5.2 – Parâmetros complementares do Gen3 para os testes de transmissão. . . . .	94
Tabela 5.3 – Parâmetros do MC-LMAC para os testes de transmissão. . . . .	94
Tabela 5.4 – Parâmetros adicionais do teste de mobilidade na topologia 9F1. . . . .	111
Tabela 5.5 – Parâmetros adicionais do teste de mobilidade na topologia 9F11. . . . .	115
Tabela 5.6 – Resultados de vazão e <i>handoff</i> obtidos na topologia 9F11. . . . .	116
Tabela 5.7 – Parâmetros de consumo de energia do rádio nas simulações. . . . .	120
Tabela 5.8 – Parâmetros complementares do Gen3 para os testes de consumo energético. . . . .	120

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>11</b>
1.1	Motivação	13
1.2	Definição do problema	14
1.3	Objetivos	14
1.4	Solução proposta	15
1.5	Contribuições	16
1.6	Estrutura do trabalho	16
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>18</b>
2.1	Redes de sensores sem fio - conceitos e técnicas	18
2.1.1	Arquiteturas de comunicação	19
2.2	Técnicas aplicadas em protocolos MAC	21
2.2.1	Protocolos TDMA	21
2.2.2	Sincronização	22
2.2.3	Multicanal	25
2.3	Multiplexação estatística	28
2.4	Métricas	29
2.5	Trabalhos relacionados	31
2.5.1	RTH-MAC	31
2.5.2	i-MAC	31
2.5.3	PriorityMAC	32
2.5.4	Z-MAC	33
2.5.5	MC-LMAC	33
2.5.6	TreeMAC	34
2.5.7	DynMAC	35
2.5.8	MCAS-MAC	35
2.5.9	Mobilidade	36
2.5.10	MA-MAC	37
2.5.11	M-TDMA	37
2.5.12	Sumário dos trabalhos analisados	37
2.6	Considerações finais	38
<b>3</b>	<b>DESCRIÇÃO DO PROTOCOLO GEN3</b>	<b>40</b>

3.1	<i>Alocação de frames e duração de ciclos</i> . . . . .	42
3.2	<i>Requisição e devolução de frames</i> . . . . .	44
3.3	<i>Configuração de slots e canais</i> . . . . .	46
3.4	<i>Pacote de controle</i> . . . . .	50
3.5	<i>Endereçamento e encaminhamento</i> . . . . .	52
3.6	<i>Associação</i> . . . . .	54
3.7	<i>Desassociação</i> . . . . .	57
3.8	<i>Handoff e buffer</i> . . . . .	60
3.9	<i>Transmissão de dados</i> . . . . .	60
3.10	<i>Multiplexação estatística coordenada - MXC</i> . . . . .	63
3.11	<i>Premissas e limitações</i> . . . . .	65
3.12	<i>Considerações finais</i> . . . . .	67
4	<b>METODOLOGIA DE AVALIAÇÃO</b> . . . . .	68
4.1	<i>Parâmetros de transmissão do protocolo</i> . . . . .	68
4.2	<i>Topologias</i> . . . . .	68
4.2.1	<i>Topologias estáticas</i> . . . . .	69
4.2.2	<i>Topologias mistas</i> . . . . .	70
4.3	<i>Sumário das topologias de teste</i> . . . . .	72
4.4	<i>Padrões de tráfego</i> . . . . .	72
4.5	<i>Cenários e métricas</i> . . . . .	73
4.6	<i>Protocolo para comparação</i> . . . . .	76
4.6.1	<i>Descrição do MC-LMAC e modificações realizadas</i> . . . . .	76
4.7	<i>Considerações finais</i> . . . . .	81
5	<b>RESULTADOS E DISCUSSÕES</b> . . . . .	83
5.1	<i>Tempo de convergência da rede</i> . . . . .	83
5.1.1	<i>Tempo de associação</i> . . . . .	83
5.1.2	<i>Tempo de estabilização da rede</i> . . . . .	89
5.1.3	<i>Considerações finais</i> . . . . .	93
5.2	<i>Transmissão de dados</i> . . . . .	94
5.2.1	<i>Taxa de entrega - CBR</i> . . . . .	95
5.2.2	<i>Vazão da rede - CBR</i> . . . . .	99
5.2.3	<i>Latência - CBR</i> . . . . .	101

<b>5.2.4</b>	<b>Taxa de entrega - VBR</b> . . . . .	105
<b>5.2.5</b>	<b>Vazão da rede - VBR</b> . . . . .	106
<b>5.2.6</b>	<b>Latência - VBR</b> . . . . .	107
<b>5.2.7</b>	<b>Considerações finais</b> . . . . .	108
<b>5.3</b>	<b>Mobilidade</b> . . . . .	111
<b>5.3.1</b>	<b>Topologia 9F1</b> . . . . .	111
<b>5.3.2</b>	<b>Topologia 9F11</b> . . . . .	114
<b>5.3.3</b>	<b>Considerações finais</b> . . . . .	117
<b>5.4</b>	<b>Consumo de energia</b> . . . . .	119
<b>5.4.1</b>	<b>Considerações finais</b> . . . . .	123
<b>6</b>	<b>CONCLUSÕES</b> . . . . .	125
<b>6.1</b>	<b>Trabalhos futuros</b> . . . . .	126
	<b>REFERÊNCIAS</b> . . . . .	128

## 1 INTRODUÇÃO

As Redes de Sensores Sem Fio (RSSF) surgiram com propostas de aplicação nas mais diversas áreas, desde monitoramento de condições ambientais, passando por aplicações na área da saúde, até aplicações militares avançadas (AKYILDIZ et al., 2002). Várias destas propostas eram voltadas para aplicações ambientais, como rastreamento de animais, detecção de incêndios florestais, mapeamento de biodiversidade e afins, como forma de prover conectividade constante em sistemas nos quais a coleta de dados era pouco automatizada. Em tais aplicações o foco principal era em economia de energia, de forma a aumentar a vida útil das redes, compostas por nós alimentados por baterias limitadas (KARL; WILLIG, 2005).

No contexto dessas propostas, os dispositivos de *hardware* disponíveis eram limitados, com baixo poder de processamento, memória, e transceptores de rádio monocanal e de baixo alcance. Com isso, as propostas de uso de RSSF eram majoritariamente relacionadas a monitoramentos ambientais, com poucas propostas considerando requisitos de aplicações de tempo real e de alta disponibilidade (ROMER; MATTERN, 2004).

Com a evolução da microeletrônica e dos dispositivos disponíveis, soluções baseadas em RSSF para indústria começaram a ser estudadas e padrões industriais como o WirelessHART e o ISA100.11a começaram a surgir (SONG et al., 2008; PETERSEN; CARLSEN, 2011). Contudo, os documentos com as especificações destes padrões são pagos, como em todo padrão proprietário, e as soluções demandam equipamentos e recursos de *hardware* específicos, além de validação dos produtos desenvolvidos pelas fundações HART e ISA.

Soluções com especificação aberta dos protocolos, bem como métodos bem definidos de avaliação e validação de equipamentos, facilitam o desenvolvimento e a certificação de novos dispositivos, o que permite ainda a redução dos custos dos equipamentos e a introdução de novas funcionalidades pelos fabricantes. Com isso, essas soluções abertas podem viabilizar uma maior penetração das RSSF no mercado de redes industriais, com o desenvolvimento independente e colaborativo de novas funcionalidades.

Em aplicações industriais os requisitos para as RSSF são diferentes. Em sistemas de controle o foco principal é na entrega de pacotes em tempo limite, garantia de entrega, confiabilidade e alta disponibilidade, ao custo de um maior consumo de energia dos nós. Em sistemas supervisórios, em que os dados coletados são apenas para acompanhamento e não há comandos de controle críticos trafegando pela rede, ainda há necessidade de entregar dados em tempo li-

mite, mas mantendo o consumo de energia reduzido (SOMAPPA; OVSTHUS; KRISTENSEN, 2014).

Para o correto funcionamento da RSSF, os protocolos de comunicação utilizados pelos nós devem ter características alinhadas aos objetivos da rede. De acordo com Suriyachai, Roedig e Scott (2012), o projeto de uma RSSF muitas vezes é iniciado com a definição do protocolo de controle de acesso ao meio (MAC - *Medium Access Control*), pois ele irá determinar o comportamento da rede em relação à capacidade de entrega de dados (latência, vazão e taxa de entrega), bem como o consumo de energia dos nós.

Os protocolos MAC podem ser divididos de acordo com seu modo de acesso em duas grandes categorias: contenção (CBP - *Contention-Based Protocol*) e escalonados (TDMA - *Time Division Multiple Access*)(KARL; WILLIG, 2005). Nos protocolos CBP os nós acessam o meio quando há algum dado a ser enviado. O controle de colisões pode ser feito de diversas formas, dentre elas: pacotes de controle, escuta e *backoff* ou detecção de preâmbulos. Em protocolos TDMA são definidos tempos específicos para que cada nó acesse o meio para transmitir ou receber dados. A duração desse acesso é determinada pelo tempo de um *slot*, e ao fim do *slot* o nó deve aguardar uma nova oportunidade de transmissão de acordo com o escalonamento da rede.

Em geral, os protocolos CBP são capazes de transmitir dados com menor latência em relação aos protocolos TDMA quando o tráfego na rede é baixo (KARL; WILLIG, 2005; RHEE et al., 2008). No entanto, com o aumento na geração de tráfego e a consequente disputa pelo meio de acesso, protocolos CBP tendem a ter operação imprevisível, com maiores taxas de perda e maior latência que os protocolos TDMA.

Por outro lado, apesar de terem comportamento previsível quando há alto tráfego na rede, nos protocolos TDMA os nós se comunicam apenas nos períodos de tempo em que foram alocados. Com isso, podem ocorrer situações em que um nó gerando muitos pacotes não consegue dar vazão suficiente aos seus dados durante seu *slot*, enquanto outro nó gerando pouco tráfego não utiliza completamente seu *slot*. Outra desvantagem dos protocolos TDMA sobre os CBP é a escalabilidade, que em alguns casos é limitada pela precisão do protocolo de sincronização de tempo, e em outros casos pela quantidade de informações que devem ser trocadas para a manutenção do escalonamento.

Além das diferenças de comportamento dos tipos de protocolo em relação a tráfego, há diferença também na forma como se adaptam a mudanças de topologia. Os protocolos TDMA

precisam manter controle sobre a alocação de *slots* e sincronização da rede, o que os torna menos adaptáveis a mudanças de topologia e mobilidade de nós. Os protocolos CBP se adaptam mais rapidamente a essas variações por necessitarem de menos controle sobre a topologia da rede.

Alguns protocolos combinam os dois modos de acesso, sendo definidos como híbridos. Em geral, esses protocolos alteram seu modo de operação de acordo com condições da rede como mobilidade, tráfego e nível de contenção. Um dos desafios dessa abordagem é a definição de quando chavear entre os modos de operação. Além disso, protocolos híbridos podem requerer intensa troca de mensagens para manutenção do escalonamento do modo TDMA, e também requerer pacotes de controle para coordenar a comunicação em modo CBP (RHEE et al., 2008).

## 1.1 Motivação

A introdução de RSSF em ambientes industriais e o seu uso para controle de processos críticos e aplicações multimídia demanda a criação de protocolos que permitam a operação da rede de forma determinística, tal que as comunicações através da rede ocorram de forma confiável e com latência previsível (AKYILDIZ; KASIMOGLU, 2004; AKYILDIZ; MELODIA; CHOWDHURY, 2007).

Para esse tipo de RSSF, fica evidente a inclinação para a escolha de um protocolo baseado em TDMA, ainda que o desempenho seja inferior em caso de baixo tráfego. Os autores Huang et al. (2013) citam como desafios para protocolos TDMA a melhoria da utilização dos *slots* em baixo tráfego, além de colocar como um foco para futuros protocolos MAC a combinação de TDMA e múltiplos canais para permitir mais comunicações simultâneas e aumentar a vazão das redes.

Além de previsibilidade na operação da RSSF, em algumas aplicações é interessante ter suporte à mobilidade de nós, mesmo que limitada (RODENAS-HERRAIZ et al., 2013). Veículos que trafegam por uma planta industrial, ou embalagens preparadas para transporte podem ter nós sensores que monitoram as condições de temperatura, umidade, vibração, dentre outras, e transmitem os dados para um sistema que permite o acompanhamento em tempo real dos produtos (SURIYACHAI; ROEDIG; SCOTT, 2012).



## 1.2 Definição do problema

Em alguns tipos de redes, como as de aplicações industriais e de monitoramentos críticos, os dados devem ser transmitidos com o menor atraso possível entre sua coleta e a entrega a um sistema de controle. Com o foco na eficiência energética, os primeiros protocolos propostos para RSSF não eram apropriados a estes tipos de aplicações.

Recentemente, novos protocolos têm sido propostos para suportar aplicações críticas, afirmando oferecer altas taxas de entrega, alta capacidade de vazão de informações e baixa latência na entrega de pacotes. No entanto, a maioria das propostas depende de interferência humana para a definição do escalonamento, limitando a capacidade de reorganização da rede e o suporte à mobilidade de nós.

Com o foco na melhoria do desempenho dessas redes e com a evolução dos transceptores de rádio, o uso simultâneo de múltiplos canais de comunicação passou a ser empregado nos protocolos MAC para RSSF. O uso de múltiplos canais introduz novos desafios de coordenação de comunicação, e a alocação de canais em alguns casos pode depender de heurísticas com alto custo computacional.

Considerando a importância da previsibilidade nas redes industriais e a maior complexidade para se coordenar comunicações com múltiplos canais, o uso de protocolos TDMA se mostra uma solução viável. Como as transmissões ocorrem em tempos bem definidos a coordenação de canais é facilitada, além de evitar a degradação das comunicações devido ao excesso de contenção em situações de alta intensidade de tráfego.

Protocolos TDMA podem subutilizar o meio de transmissão em situações de tráfego reduzido ou desbalanceado entre os nós. Como geralmente esses protocolos determinam períodos fixos de tempo para que os nós transmitam dados, é comum que tanto os nós gerando tráfego intenso quanto os nós gerando baixo tráfego possuam a mesma quantidade de tempo para realizar essas transmissões. Esse é um fenômeno indesejado, pois o nó que está com maior demanda pelo meio de transmissão terá sua operação interrompida em favor de um nó que tem baixa demanda.

## 1.3 Objetivos

O objetivo deste trabalho é o desenvolvimento de um protocolo MAC que permita a transmissão de dados por uma rede multi-saltos com alta taxa de entrega, alta vazão de dados

e baixa latência. O protocolo deve apresentar operação estável quando submetido a diferentes intensidades de tráfego, de modo que seja viável para aplicações críticas.

O protocolo deve ser autoconfigurável, definindo sua quantidade de *slots* e seu escalonamento sem a necessidade de intervenção ou conhecimento prévio das condições da rede, permitindo a associação de todos os nós que estiverem na área de cobertura da rede. Estendendo esse objetivo, o protocolo deve ainda suportar a mobilidade de nós, mesmo que a velocidades reduzidas, causando impacto mínimo na transmissão dos dados.

Para aumentar a utilização do meio de transmissão quando a geração de tráfego dos nós é desbalanceada, o protocolo deve empregar um mecanismo de multiplexação estatística. Esse mecanismo deve permitir o uso oportunístico dos *slots* de tempo subutilizados, dando mais oportunidades de comunicação para os nós que têm mais demanda pelo meio de transmissão.

#### **1.4 Solução proposta**

Este trabalho apresenta o Gen3, um protocolo MAC voltado para redes que executam aplicações que dependem de altas taxas de entrega, alta vazão de dados e latência previsível. O protocolo é baseado em TDMA, permitindo o suporte a cargas intensas de tráfego, sem a ocorrência de colisões no meio de transmissão. O Gen3 ainda emprega múltiplos canais de comunicação, permitindo a ocorrência de transmissões simultâneas livres de interferência, em canais distintos.

O Gen3 configura a rede automaticamente, sem a necessidade de conhecimento prévio da disposição dos nós em termos de densidade ou organização, requerendo apenas que seja possível formar uma rede conexa. A entrada de novos nós faz com que mais *slots* sejam adicionados à rede, assim como a saída de nós implica na redução da quantidade de *slots*, permitindo que a rede opere de forma eficiente de acordo com as condições momentâneas.

A alocação dos *slots* para os nós, bem como a definição dos canais de comunicação utiliza algoritmos simplificados, que não exigem esforço computacional considerável dos nós. Todas as informações de controle são trocadas com pacotes de tamanho fixo, sendo a precisão do protocolo de sincronização o principal fator limitante da escalabilidade do Gen3.

Por meio da multiplexação estatística, uma técnica pouco explorada em RSSF, o Gen3 permite o uso oportunístico de *slots* de tempo subutilizados, visando melhorar seu desempenho em condições de tráfego intenso ou desbalanceado entre os nós.

## 1.5 Contribuições

Como contribuição direta relacionada a este trabalho, foi publicado no *XXXIV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC*, o trabalho intitulado *LMT-MAC: Um Protocolo MAC Multicanal Livre de Colisões Para Redes de Sensores Sem Fio* (JÚNIOR; MARQUES; CORREIA, 2016). Esse trabalho introduziu o *Lightweight Multichannel Tree-MAC*, um protocolo MAC TDMA que forneceu as bases para o desenvolvimento do Gen3.

O LMT-MAC depende de configuração manual dos nós, utiliza um esquema simplificado de alocação de canais, não emprega multiplexação estatística e não suporta variações de topologia ou mobilidade de nós. Apesar disso limitar a adaptabilidade do protocolo, a menor complexidade de sua implementação permite que seja usado em nós com recursos limitados de *hardware*, tendo sido avaliado tanto por simulações, quanto por implementação em nós reais.

Os resultados obtidos com o LMT-MAC e apresentados no trabalho fundamentaram várias decisões do projeto do Gen3, de modo que este tenha sido desenvolvido considerando os requisitos e limitações de dispositivos reais.

Como contribuição indireta, foi publicado no *Twenty-First IEEE Symposium on Computers and Communications - ISCC*, o trabalho intitulado *ISCRa - An Intelligent Sensing Protocol for Cognitive Radio* (MARQUES et al., 2016). O ISCRa é um protocolo de sensoriamento inteligente para rádios cognitivos que opera na camada MAC, permitindo a alocação dinâmica do espectro e acesso oportunístico em frequências alocadas a usuários licenciados quando estiverem temporariamente ociosas.

## 1.6 Estrutura do trabalho

O Capítulo 2 apresenta um referencial teórico com os principais conceitos abordados neste trabalho. Inicialmente são apresentadas as Redes de Sensores Sem Fio e como elas são geralmente organizadas. Em seguida, com o foco do trabalho nos protocolos de controle de acesso ao meio, são discutidas diferentes técnicas utilizadas nesses protocolos. O capítulo é encerrado com o levantamento e a análise dos trabalhos relacionados, apresentando os principais pontos de cada um.

O Gen3 é apresentado em detalhes no Capítulo 3. Inicialmente é definida uma notação para auxiliar essa descrição. Os formatos de divisão de tempo, alocação de *slots* e de canais

são apresentados, bem como os formatos de pacotes para controle e transmissão de dados. A descrição traz exemplos para melhor compreensão do funcionamento da associação de novos nós na rede e da multiplexação estatística, dado que são elementos cruciais da operação do protocolo.

O Capítulo 4 apresenta a metodologia utilizada para a avaliação do Gen3 e sua comparação com outras soluções. As topologias usadas nos testes, assim como os modelos de movimento e padrões de tráfego são detalhados. O capítulo ainda traz as métricas definidas para avaliação, bem como a escolha do protocolo de comparação, sua implementação e as modificações realizadas.

Os resultados obtidos com os testes são apresentados e discutidos no Capítulo 5. No capítulo são avaliadas tanto as métricas inerentes ao próprio protocolo, como o tempo de associação dos nós e a configuração completa da rede, quanto os resultados da transmissão de dados, relativos a taxa de entrega, vazão e latência, comparados a outro protocolo.

Por fim, o Capítulo 6 apresenta as conclusões obtidas com este trabalho, os pontos positivos e as limitações da solução desenvolvida, bem como propostas para a evolução do protocolo em trabalhos futuros.

## 2 REFERENCIAL TEÓRICO

Neste capítulo é apresentado o referencial teórico que enfoca os conceitos utilizados para a elaboração deste trabalho.

### 2.1 Redes de sensores sem fio - conceitos e técnicas

Redes de Sensores Sem Fio (RSSF) são redes compostas por vários nós que operam conjuntamente para a execução de uma determinada aplicação. Os nós sensores se comunicam por meio de transceptores de rádio, sendo uma solução flexível para cobertura e monitoramento em diversos ambientes (AKYILDIZ et al., 2002).

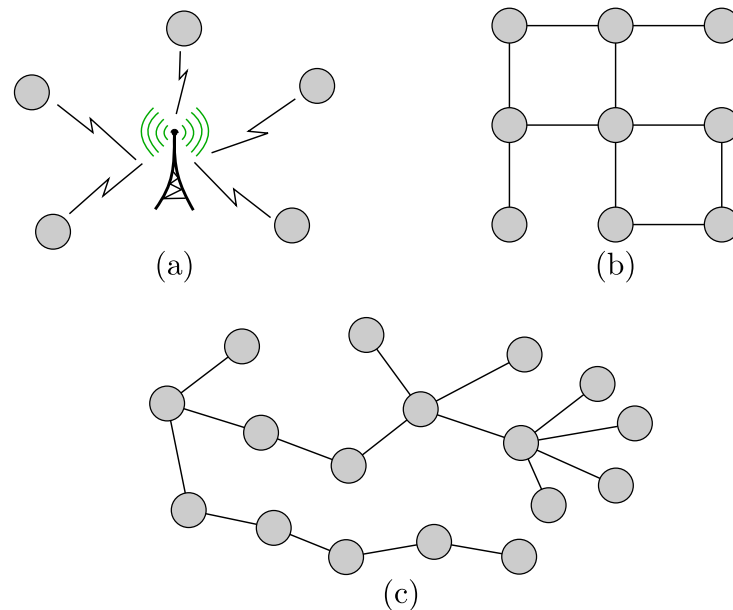
Recentemente as RSSF têm sido propostas para aplicações industriais, nas quais a troca de mensagens de forma rápida e confiável é o foco principal (SURIYACHAI; ROEDIG; SCOTT, 2012). Para permitir isso, protocolos e técnicas têm sido estudados e desenvolvidos de modo a permitir a operação confiável e eficiente dessas redes.

Diferentemente de redes infraestruturadas, nas quais os nós se conectam a um ponto de acesso que oferece cobertura a uma área, nas RSSF os nós sensores podem fornecer conectividade aos demais, formando redes ad-hoc multi-saltos que estendem essa área de cobertura. Essa diferença conceitual implica também em mudanças nos protocolos empregados por essas redes, tal que operem de forma eficiente de acordo com as aplicações em que são utilizadas.

A topologia física, ou disposição dos nós em RSSF, também varia com a aplicação, podendo ser regular, com os nós dispostos de forma organizada em um espaço, ou irregular, na qual os nós são dispostos de forma aleatória (RUIZ et al., 2004). A topologia lógica da rede define como os nós interagem e como se dão as comunicações entre eles, tendo como exemplo topologias em árvore ou *mesh*.

Na Figura 2.1a é ilustrada uma topologia infraestruturada, na qual um ponto fixo fornece conectividade aos nós, na Figura 2.1b uma rede disposta regularmente com topologia lógica em *mesh*, cujas conexões são indicadas pelas linhas. Na Figura 2.1c é ilustrada uma rede com disposição irregular, com topologia lógica em árvore. Cada forma de organização lógica tem suas vantagens e desvantagens relativas a roteamento de mensagens, adaptabilidade a mudanças, formação da rede dentre outros fatores que devem ser levados em conta no projeto dos protocolos.

Figura 2.1 – Tipos de topologia



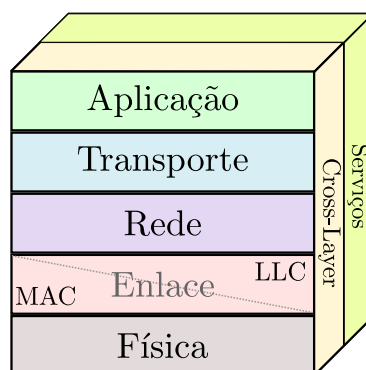
Legenda: (a) Topologia infraestruturada; (b) Topologia física regular e topologia lógica em *mesh*; (c) Topologia física irregular e lógica em árvore.

Fonte: do autor (2017).

### 2.1.1 Arquiteturas de comunicação

Em redes de computadores a organização dos protocolos é geralmente feita em camadas, cada uma oferecendo serviços para as camadas superiores (TANENBAUM, 2002). Dois modelos de referência são utilizados para definir a divisão e as atribuições de cada camada, sendo eles o modelo OSI (*Open Systems Interconnection*) padronizado pela ISO (*International Standards Organization*) (DAY; ZIMMERMANN, 1983; DAY, 1995), e o modelo TCP/IP, uma arquitetura desenvolvida pelo Departamento de Defesa dos EUA, que foi definido posteriormente como um modelo de referência (TANENBAUM; WETHERALL, 2011). Neste trabalho é seguido um modelo híbrido definido por Tanenbaum e Wetherall (2011), com as camadas: Aplicação, Transporte, Rede, Enlace e Física.

As RSSF empregam um modelo *cross-layer* (RUIZ et al., 2004), ilustrado na Figura 2.2, que visa otimizar o uso dos recursos dos nós sensores ao permitir a comunicação de camadas não adjacentes, diferente dos modelos de referência (KARL; WILLIG, 2005). Além disso, alguns recursos podem não ser implementados por serem desnecessários em determinada RSSF ou incorporados de forma simplificada em outras camadas, reduzindo a complexidade dos projetos ao custo de uma menor padronização entre as redes.

Figura 2.2 – Organização em camadas com serviços *cross-layer*.

Fonte: do autor (2017).

Por ter maior enfoque na camada MAC, os serviços e as características das camadas de aplicação, transporte e rede não serão discutidos em detalhes. Nessa organização em camadas, a camada de aplicação engloba os programas e serviços que operam sobre a infraestrutura da rede, executando aplicações de interesse dos usuários e companhias. A camada de transporte provê comunicação entre processos sendo executados em diferentes nós. A camada de rede provê roteamento e encaminhamento de pacotes através dos vários nós da rede, compreendendo também funções de endereçamento.

A camada de enlace é composta pelas subcamadas: LLC - *Logical Link Control* (controle lógico da conexão) e MAC - *Medium Access Control* (controle de acesso ao meio). O protocolo desenvolvido neste trabalho se encontra na subcamada MAC, que controla o acesso ao meio de transmissão. Esse controle é realizado em termos de: tempo que o nó pode se manter transmitindo pacotes; tempo que o nó escuta o meio para receber pacotes; a frequência que configurará o transceptor; além de outras operações necessárias para evitar a ocorrência de colisões e permitir a troca de pacotes entre os nós.

Os serviços da camada de enlace operam sobre os serviços da camada física, que trata da transmissão de bits por um canal de comunicação. No caso das RSSF utilizando radio-frequência, os dados são convertidos em sinais eletromagnéticos que são transmitidos em uma determinada frequência. No receptor esses sinais são identificados, decodificados e convertidos novamente em dados, os quais podem ser interpretados pelas camadas superiores.

## 2.2 Técnicas aplicadas em protocolos MAC

Por utilizar um meio compartilhado, as transmissões de um nó podem ser identificadas por todos os outros em seu raio de alcance. Se um nó detectar duas ou mais transmissões ao mesmo tempo e no mesmo canal, ele não consegue distinguir as transmissões e extrair as informações. Esse fenômeno é chamado de colisão, e a principal função do protocolo MAC é coordenar as comunicações para que colisões não ocorram (BACHIR et al., 2010).

Como mencionado no Capítulo 1, os protocolos de acesso ao meio podem ser classificados como CBP, TDMA ou híbridos, cada um com características próprias e diferentes mecanismos para controle e prevenção de colisões.

### 2.2.1 Protocolos TDMA

Um dos principais métodos utilizados para tentar evitar a ocorrência de colisões é o uso de TDMA, no qual um período de tempo é dividido em fatias que são alocadas aos nós. Cada nó sabe exatamente em qual momento deve ativar seu rádio para a transmissão ou recepção de dados, podendo desativá-lo para economizar energia quando estiver fora desse período.

Esses protocolos têm natureza cíclica, dado que as oportunidades de transmissão devem se repetir ao longo do tempo. Dessa forma, o tempo é primeiramente dividido em uma unidade maior, dentro da qual todo nó tem pelo menos uma oportunidade de comunicação com algum vizinho. Essa unidade recebe diferentes nomes nos diferentes protocolos, podendo ser chamada de *Frame*, *Superframe* ou *Ciclo* (O'DONOVAN et al., 2013; ABDELI; ZELIT; MOUSSAOUI, 2013; SONG et al., 2009).

Essa unidade é subdividida em unidades menores, como *slots* ou *frames*, que são alocadas aos nós participantes da rede. A quantidade de *slots* pode ser fixa ou variável, e alguns podem ser deixados livres (sem nó alocado) de acordo com mecanismos e propósitos dos protocolos. Indo além, essas unidades ainda podem ser subdivididas em outras menores e empregadas a finalidades diferentes que variam com os protocolos.

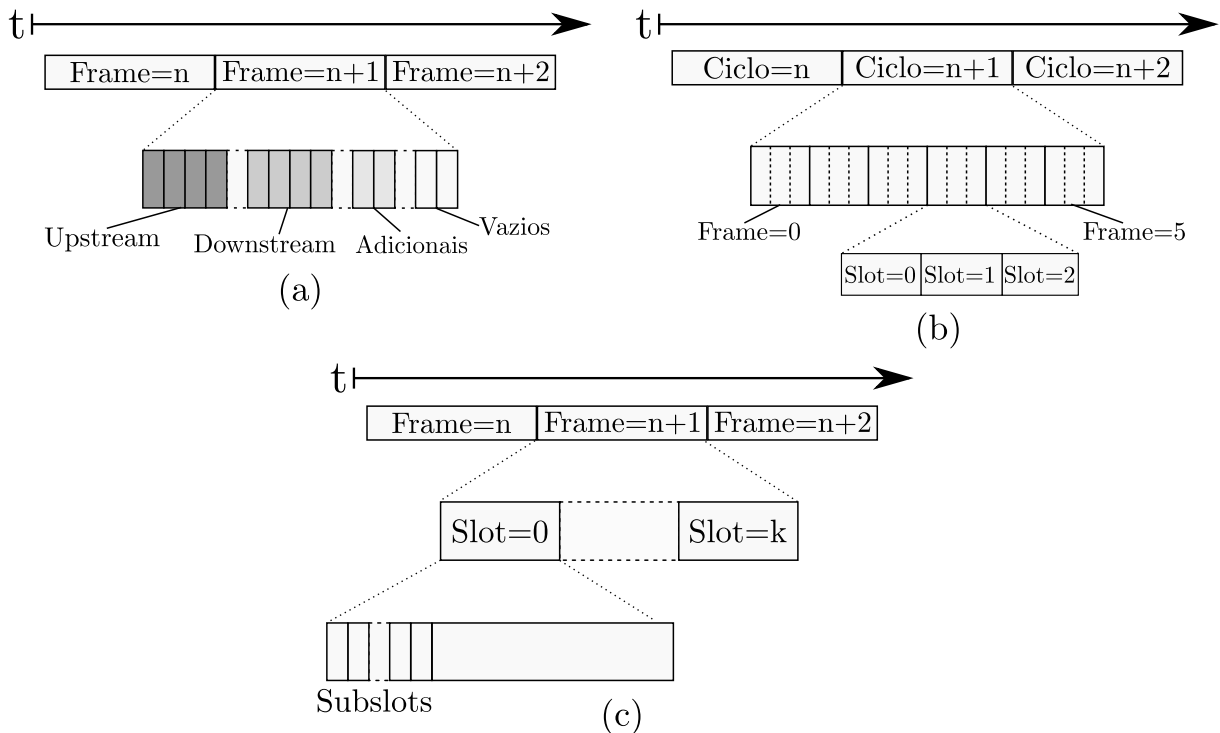
A Figura 2.3 ilustra três formatos de divisão comumente encontrados em protocolos MAC TDMA. Na Figura 2.3a é mostrado o esquema utilizado pelo GinMAC (SURIYACHAI; BROWN; ROEDIG, 2010), com o tempo dividido em *frames*, composto por vários *slots* com objetivos bem definidos, *upstream*, *downstream*, adicionais e vazios.

Na Figura 2.3b é ilustrado o formato de divisão do TreeMAC (SONG et al., 2009), no qual o tempo é dividido em ciclos, cada ciclo composto por vários *frames* e cada *frame* contendo



uma quantidade fixa de 3 *slots*. Por fim, na Figura 2.3c é mostrado o esquema utilizado no MC-LMAC (INCEL et al., 2011), no qual um *frame* é dividido em vários *slots* e cada *slot* é dividido em *subslots*. O protocolo proposto neste trabalho adota um esquema baseado no da Figura 2.3b, com *ciclos*, *frames* e *slots*.

Figura 2.3 – Tipos de divisão de tempo em protocolos TDMA.



Legenda: (a) Divisão em *frames* e *slots*; (b) Divisão em *ciclos*, *frames* e *slots*; (c) Divisão em *frames*, *slots* e *subslots*.

Fonte: (a) Adaptado de (SURIYACHAI; BROWN; ROEDIG, 2010); (b) Adaptado de (SONG et al., 2009); (c) Adaptado de (INCEL et al., 2011).

Independente do tipo de divisão utilizado, os nós participantes da rede devem seguir o escalonamento corretamente e operar apenas nos momentos estabelecidos pelo protocolo. Caso contrário, o comportamento da rede é imprevisível e pode ser necessário refazer a configuração de todos os nós.

## 2.2.2 Sincronização

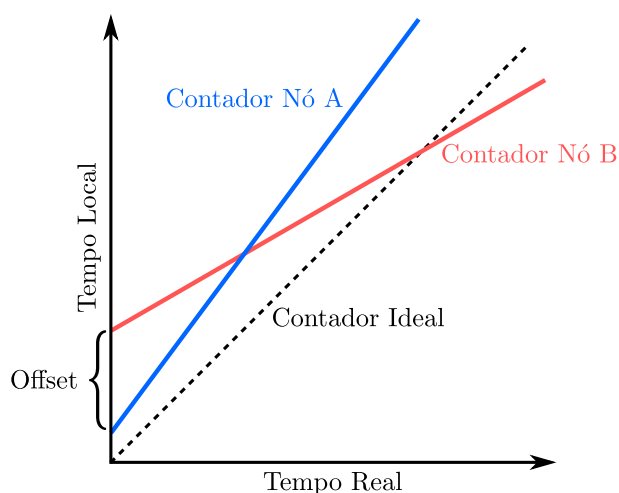
Para que o acompanhamento do escalonamento dos protocolos TDMA seja feito corretamente, é necessária a sincronização dos nós. O primeiro e principal elemento a ser sincronizado entre os nós é o tempo. A maioria dos microcontroladores possuem contadores internos com

precisão de microssegundos, que são incrementados desde que o microcontrolador é ligado, com base no sinal do oscilador.

Idealmente esse contador seria incrementado na mesma velocidade em todos os nós, mas devido a imperfeições dos osciladores existe uma variação no tempo marcado entre os diferentes nós e em relação ao tempo real (WU; CHAUDHARI; SERPEDIN, 2011). Ainda segundo Wu, Chaudhari e Serpedin (2011), osciladores comumente usados em nós para RSSF possuem variação de frequência de até 40 ppm, causando diferença de até 40  $\mu$ s a cada segundo nos contadores dos nós sensores. Além disso, os nós da RSSF não são ligados ao mesmo tempo, gerando uma diferença inicial nesses valores.

A Figura 2.4 ilustra o comportamento dos contadores de dois nós em relação a um contador ideal. É possível notar que em um contador ideal o crescimento do tempo local, marcado pelo nó, é igual ao crescimento do tempo real, podendo ser modelado como uma equação da reta  $y = mx + n$ , com coeficiente angular  $m$  igual a 1 e coeficiente linear  $n$  igual a 0. Quando se trata de sincronização, o coeficiente linear é comumente chamado de *offset* (deslocamento) e é relacionado à diferença inicial de tempo entre os dois contadores. Já o coeficiente angular é chamado de *skew* (inclinação) e indica o erro de crescimento desse contador em relação ao contador ideal.

Figura 2.4 – Marcação de tempo em nós sensores.



Legenda: Contador de tempo Ideal (preto) e os contadores de dois nós (A e B) com variações para o tempo ideal.

Fonte: do autor (2017).

Na Figura 2.4, o oscilador do nó A tem frequência mais alta que a do contador ideal, implicando em um crescimento mais rápido de seu valor. Por outro lado, o oscilador do nó

B está gerando frequência mais baixa que o ideal, fazendo com que seu contador tenha um crescimento mais lento.

Como não há um nó com o relógio ideal em uma RSSF, algum nó deve ser escolhido como referência para que os demais se sincronizem. Supondo que o tempo do nó A seja utilizado como referência (tempo global da rede), o nó B deve sincronizar seu contador local ao de A. Para isso A transmite pacotes de sincronização com o valor de seu contador, que é então utilizado pelos demais nós para estimar os valores de *offset* e *skew*. O protocolo de sincronização não necessariamente altera diretamente o valor do contador, mas estima a relação entre *offset* e *skew* do nó atual em relação à referência e aplica esses valores ao contador local sempre que é necessário saber o tempo global da rede.

Existem diferentes abordagens para a troca de mensagens de sincronização. No protocolo RBS - *Reference Broadcast Synchronization* (ELSON; GIROD; ESTRIN, 2002), o nó de referência transmite pacotes em *broadcast*. Assumindo que essas mensagens chegarão aos receptores no raio de alcance ao mesmo tempo, esses receptores posteriormente trocam mensagens aos pares para calcular o *offset* e o *skew* (que no RBS é calculado utilizando o método de mínimos quadrados).

Já o LTS - *Lightweight Tree-based Synchronization* (GREUNEN; RABAEY, 2003) utiliza trocas de mensagens diretamente entre pares de nós, permitindo a sincronização com a troca de apenas três mensagens entre esse par. O FTSP - *Flooding Time Synchronization Protocol* (MARÓTI et al., 2004) permite a sincronização de nós em relação a uma referência, como o LTS, mas transmite os pacotes de sincronização em *broadcast* como o RBS. O FTSP estima os valores de *offset* e *skew* por regressão linear sobre uma quantidade predefinida de amostras.

Em redes multi-saltos, em que nem todos os nós são capazes de ouvir o *broadcast* do nó de referência, os demais nós já sincronizados oferecem sincronização aos demais no seu raio de alcance, transmitindo nos pacotes de sincronização uma estimativa de tempo global feita por eles. Essa abordagem, no entanto, faz crescer o erro de sincronização a cada salto, sendo uma política padrão utilizar o tempo global do nó mais próximo (em saltos) do nó de referência (MARÓTI et al., 2004).

No LTS, em que há um crescimento linear desse erro por salto, a precisão obtida pelos autores foi de aproximadamente 1 ms em um nó a 15 saltos da referência imediatamente após a sua sincronização. Esse valor pode crescer devido às variações de frequência do oscilador, que afetam a velocidade em que os contadores dos nós são incrementados.

Devido à variação da velocidade de incremento dos contadores, a troca de mensagens de sincronização deve ser constante. Alguns protocolos são capazes de estimar essa velocidade de variação, permitindo que o intervalo de envio das mensagens de sincronização seja maior, além de atingirem precisão de sincronização inferior a  $100 \mu\text{s}$  (MARÓTI et al., 2004; SOMMER; WATTENHOFER, 2009).

De modo geral, nos protocolos TDMA a atualização dos parâmetros da rede é feita em função do tempo global, por isso, quanto mais precisa a estimativa, menor a chance de erros na configuração dos nós. Dessa forma, além do tempo, os valores de *slot*, *frame* e ciclo também são comumente sincronizados. Esses valores são importantes para definir as configurações e o comportamento dos nós em um determinado instante. De acordo com o valor do *slot* e o escalonamento da rede, um nó pode decidir se deve transmitir pacotes, aguardar pelas transmissões de outros nós ou até mesmo desativar seu rádio para economizar energia. Esses valores são transmitidos juntamente com o tempo global nos pacotes de controle.

Temporizadores são utilizados para informar o nó que o *slot* se encerrou. O nó então atualiza os valores dos marcadores locais, bem como suas configurações. Como a sincronização de tempo não é perfeita, após o sinal do temporizador o nó aguarda um tempo de guarda, no qual nenhuma ação é tomada. Os tempos de guarda visam garantir que todos os nós tenham se configurado corretamente após a mudança de *slot* e são maiores que o erro de sincronização esperado.

Com o esquema TDMA bem definido e uma sincronização precisa dos nós, a rede pode operar de forma estável e previsível mesmo com altas cargas de tráfego. Com o escalonamento já definido, dar suporte a mais comunicações simultâneas permite o aumento da capacidade da rede tornando as transmissões mais confiáveis e com menor atraso. A exploração dos recursos disponíveis nos transceptores de rádio atuais, como a capacidade de operar em diferentes canais, é uma opção clara para a melhoria desses protocolos.

### **2.2.3 Multicanal**

A maioria dos transceptores de rádio utilizados nos primeiros dispositivos para RSSF era capaz de operar em apenas um canal, principalmente como forma de reduzir o custo dos componentes, sem a possibilidade de mudança desse canal. Posteriormente, foram desenvolvidos transceptores que podiam ser configurados para operar em canais diferentes (INCEL, 2011). Em todos esses casos, e no decorrer deste trabalho, é assumido que os transceptores

operam apenas em modo *Half-Duplex*, sem a capacidade de operar como transmissor e receptor ao mesmo tempo.

Essa capacidade de configuração do canal de operação abre caminho para protocolos que permitam que pares diferentes de nós em uma rede se comuniquem simultaneamente e em canais distintos sem que causem interferências entre si, trazendo diversos benefícios a essas redes. Os autores Souza e Minet (2015) citam alguns benefícios esperados ao se utilizar comunicação multicanal em RSSF, dentre eles:

- Aumento da quantidade de transmissões simultâneas por meio da alocação de canais diferentes a nós próximos, permitindo comunicação livre de interferências e aumentando a vazão dessas redes.
- Aumento da taxa de entrega, principalmente de dados com alta prioridade.
- Redução do atraso na transmissão de dados coletados.
- Aumento da robustez da rede na presença de interferências e ruídos, dado que redes multicanais podem evitar canais com altos níveis de ruído.

Apesar das vantagens, comunicações multicanais introduzem desafios classificados por Incel (2011) em três categorias: sobreposição de canais, troca de canal do transceptor e coordenação de canais.

Os canais utilizados pelos transceptores nem sempre são ortogonais, logo, um determinado canal pode se sobrepor parcialmente a canais adjacentes, causando interferências e reduzindo a qualidade do sinal nos receptores. Em protocolos multicanal é sempre preferível a utilização de canais ortogonais, mas dependendo do transceptor, a quantidade de canais ortogonais pode ser baixa, implicando em redução da performance do protocolo (INCEL et al., 2006). A seleção dos canais utilizados para as comunicações deve levar em conta essas características e a relação interferência versus performance.

Outra questão a ser considerada é o tempo gasto pelo transceptor para se configurar para um canal diferente. Esse tempo varia de acordo com o transceptor e deve ser considerado na implementação dos protocolos, de modo que possam aguardar o tempo necessário para essa configuração. Nos rádios CC2420 da *Texas Instruments* esse valor é aproximadamente  $200 \mu s$  (INSTRUMENTS, 2006), enquanto nos rádios nRF24101+ da *Nordic Semiconductor* esse valor é de  $130 \mu s$  (NORDIC\_SEMICONDUCTOR, 2013).

A coordenação de canais tem maior complexidade que os desafios anteriores, os quais ainda variam com o tipo de alocação utilizado. Essa alocação pode ser: *fixa*, com os canais predefinidos, como por exemplo uma rede dividida em *clusters* em que cada *cluster* opera em um canal distinto (YU; CHANG, 2007); *semi-dinâmica*, na qual os canais são alocados aos transmissores e receptores, e os nós trocam seu canal de operação para transmitir para um determinado vizinho (JÚNIOR; MARQUES; CORREIA, 2016); ou *dinâmica*, na qual a transmissão de dados ocorre após uma fase de seleção de canal, sendo dependente de transmissões em CCC (Canal de Controle Comum) ou de varredura de canais para que ocorra a definição do canal para a troca de dados (INCEL et al., 2011).

Dentre os principais problemas envolvendo a coordenação estão:

- *Entrada de novos nós*: no caso de alocação fixa, a entrada de novos nós na rede pode atrapalhar uma configuração pré-estabelecida. No caso de alocação dinâmica e semi-dinâmica, novos nós podem ter que varrer vários canais buscando pacotes de controle para efetuar sua associação à rede. Por isso, alguns protocolos definem um Canal de Controle Comum (CCC) para facilitar a transmissão de informações de controle e configuração (MARQUES et al., 2016; INCEL et al., 2011; DIAB; CHALHOUB; MISSON, 2013).
- *Suporte a broadcasts*: como os nós trabalham em canais diferentes, o suporte a transmissões em *broadcast* fica prejudicado. Um nó no alcance do *broadcast* pode estar configurado para outro canal e não detectar a mensagem recebida.
- *Problema da surdez*: um nó A pode considerar um outro nó B fora de seu alcance se B nunca responder às transmissões de A. No entanto, B pode estar no alcance mas configurado para um canal diferente de A quando essas transmissões ocorrem. Esse tipo de problema ocorre principalmente em protocolos de contenção como o CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*), em um caso em que B não responderia às mensagens de controle de A.
- *Problema do terminal escondido multicanal*: esse problema também é relacionado ao uso de protocolos como o CSMA/CA em redes multicanal (SO; VAIDYA, 2004). Nas redes monocanal, os pacotes RTS (*Request To Send*) e CTS (*Clear To Send*) são utilizados para evitar o problema do terminal escondido. Já em multicanal, por estar configurado em outro canal um nó pode não detectar a transmissão de pacotes RTS e CTS por algum vizinho e causar colisões.

A comunicação multicanal em protocolos TDMA é facilitada em relação a protocolos CBP por já existir uma coordenação prévia do protocolo relacionada ao escalonamento de *slots*, sincronização de tempo e outras configurações. No entanto, mesmo utilizando multicanal para aumentar a quantidade de transmissões simultâneas, ainda há espaço para melhoria por meio do aproveitamento dos *slots* de tempo quando são subutilizados.

### 2.3 Multiplexação estatística

A multiplexação estatística tem o objetivo de otimizar a utilização de um recurso sujeito a padrões de uso aleatórios (CHANDRA, 2003). Técnicas de multiplexação estatística têm sido aplicadas em protocolos de telefonia há anos, tirando proveito principalmente das características de conversação entre pessoas. Um transmissor capaz de classificar sinais relativos a fala pode aproveitar os períodos de silêncio para a transmissão de dados, aumentando a utilização do canal de transmissão (ROBERGE; ADOUL, 1986).

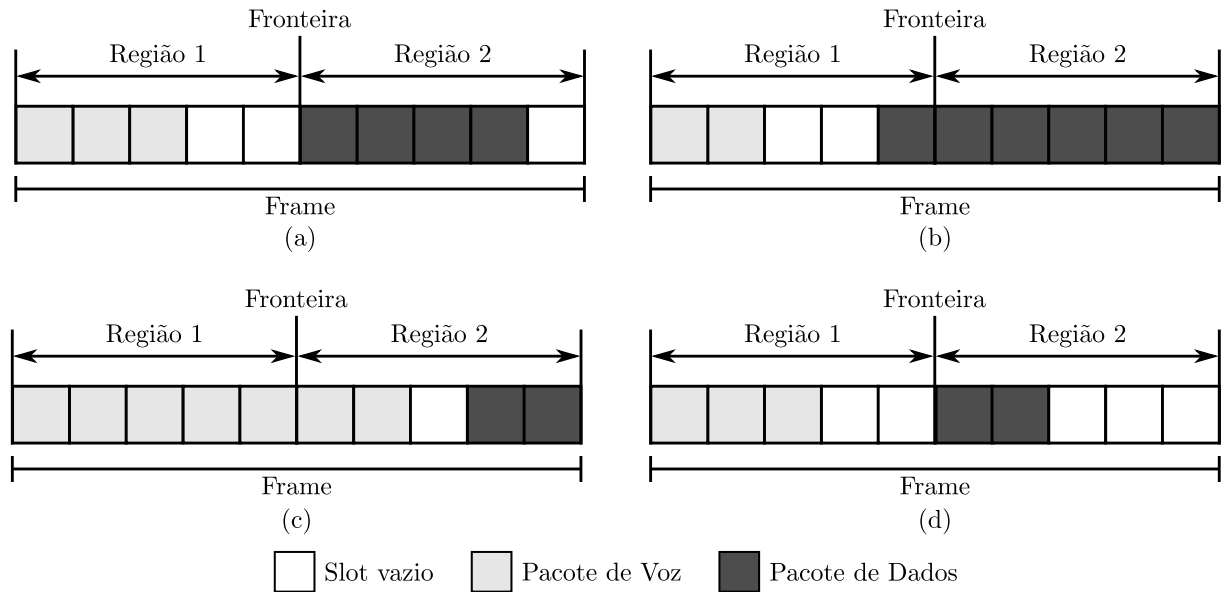
Ainda no contexto de redes de telefonia com transmissão de voz e dados, em (LEE; UN, 1985) são discutidas quatro formas de multiplexação dessas informações em um conjunto finito de *slots* de tempo: (i) Fronteira Fixa (FF); (ii) Fronteira Móvel para Dados (FMD); (iii) Fronteira Móvel para Voz (FMV); (iv) Fronteira Móvel para Voz e Dados (FMVD).

A Figura 2.5 apresenta casos de ocupação de um *frame* de acordo com o tráfego e a forma de multiplexação utilizada. O *frame* é composto por 10 *slots*, nos quais se pode transmitir um pacote de voz ou um pacote de dados, e é dividido em duas regiões, a primeira com prioridade para pacotes de voz e a segunda para pacotes de dados.

Na Figura 2.5a é ilustrada a multiplexação com FF, no qual os pacotes de voz e dados são alocados exclusivamente nas regiões 1 e 2 respectivamente. Já na Figura 2.5b, utilizando FMD, os pacotes de dados podem ser alocados também para a região 1, aproveitando os *slots* vazios. Na Figura 2.5c é utilizada a FMV, com os pacotes de voz ocupando exclusivamente a região 1 e os *frames* livres da região de dados. Por fim, na Figura 2.5d em que é utilizada a FMVD os pacotes de dados e voz podem ser alocados fora de suas regiões com prioridade secundária.

Esse conceito de alocação adaptativa de recursos pode ser estendido de outras formas. Os autores Dunlop et al. (1995) propuseram a aplicação de multiplexação estatística a um protocolo TDMA para redes de telefonia de terceira geração, permitindo a alocação de recursos durante períodos de alta atividade e sua realocação quando essa atividade se encerra.

Figura 2.5 – Multiplexação estatística em rede de telefonia e dados.



Multiplexação estatística de pacotes de voz e dados em um *frame* com 10 *slots*. (a) Fronteira Fixa; (b) Fronteira Móvel para Dados; (c) Fronteira Móvel para Voz; (d) Fronteira Móvel para Voz e Dados.

Fonte: adaptado de (LEE; UN, 1985).

Apesar de existirem diversas propostas de aplicação de multiplexação estatística em redes de telefonia, pouco se observa a aplicação desse mecanismo em RSSF. Pelo grande foco dos protocolos de RSSF em eficiência energética, esse mecanismo pode ter sido inexplorado por necessitar de escuta adicional de canal ou outros tipos de coordenação para sua implementação. No entanto, redes com foco principal em desempenho podem tirar proveito da multiplexação estatística para melhor alocação de seus recursos e aumento da capacidade.

## 2.4 Métricas

A avaliação dos protocolos deve ser feita com base em métricas bem definidas. Dentre as principais métricas a serem avaliadas estão: latência (ou atraso), vazão e taxa de entrega. De acordo com Kurose e Ross (2013), existem diversas formas de atraso em uma rede de comutação de pacotes, sendo elas:

- *Processamento*: dado pelo tempo necessário para que um nó examine um pacote e determine qual ação a ser tomada com ele.
- *Transmissão*: considerando um *buffer* em que os pacotes são transmitidos na ordem de chegada, os últimos pacotes a serem colocados em *buffer* terão um atraso maior de trans-



missão, por terem que aguardar a transmissão dos demais. Esse atraso pode ser afetado caso algum tipo de priorização de pacotes seja utilizado.

- *Propagação*: tempo necessário para que o pacote seja transmitido através do meio físico.

Combinando todas as fontes de atraso obtém-se o atraso fim-a-fim, dado pelo tempo decorrido desde a criação do pacote no nó de origem até sua chegada ao nó de destino.

Já a vazão pode ser medida pela taxa de bits por segundo em que um nó recebe dados. Em RSSF diferentes formas de avaliar a vazão são adotadas, dentre elas a vazão agregada, considerando a vazão combinada de todos os nós em um determinado instante, ou pela quantidade de pacotes chegando a um determinado nó por unidade de tempo. Com a vazão, avalia-se tanto a capacidade de transmissão dos transceptores, quanto a capacidade dos protocolos utilizados em entregar dados de forma eficiente.

Os dispositivos de rede possuem *buffers* para armazenamento dos pacotes antes de sua transmissão. Esses *buffers* possuem tamanho limitado e quando são esgotados ocorre o descarte de pacotes, seguindo algum tipo de política (como descarte dos pacotes mais antigos ou com menor prioridade). Além disso, devido a interferências ou outros fatores, a transmissão de um pacote para seu destino pode ser mal sucedida.

Os protocolos da camada de enlace são responsáveis pela entrega de dados ponto-a-ponto (entre pares de nós que se comunicam diretamente), ficando a cargo das camadas superiores a garantia de entrega fim-a-fim (através de toda a rede). Dessa forma, a responsabilidade por garantir a entrega das informações ao destino fica a cargo dos protocolos de camadas superiores, cabendo ao enlace adotar mecanismos simplificados para recuperação de erros. Ainda assim, os protocolos da camada de enlace devem buscar reduzir as taxas de erros e perdas de pacote, caso contrário, o desempenho geral da rede fica prejudicado e mais tráfego relacionado a retransmissões é gerado.

Além dessas métricas, quando se trata de redes em que há mobilidade dos nós, deve-se analisar o tempo de *handoff*. O *handoff* considera o tempo entre a desconexão de um nó em determinado ponto e sua reconexão em outro ponto da rede. Enquanto os nós estão desconectados da rede, o protocolo de enlace deve armazenar os pacotes recebidos da camada superior em um *buffer*, até que este nó se reconecte à rede e possa transmitir pacotes novamente. Caso o processo de reentrada na rede tome muito tempo, pode ocorrer o esgotamento de *buffer* e o consequente descarte de pacotes.

## 2.5 Trabalhos relacionados

Esta seção traz uma seleção de protocolos MAC que têm por objetivo a transmissão de dados com baixa latência, alta vazão e alta taxa de entrega, com potencial de uso em aplicações industriais ou críticas. O principal foco está em protocolos TDMA multicanais e também em protocolos que tenham a capacidade de se adaptar a variações de topologia e mobilidade de nós.

### 2.5.1 RTH-MAC

O protocolo *Real Time Hybrid MAC* (RTH-MAC) (ABDELI; ZELIT; MOUSSAOUI, 2013) emprega um esquema TDMA centralizado, configurado por uma estação base, com foco em obter baixa latência. Após uma fase inicial de descoberta de vizinhos, a estação base executa um algoritmo para alocação de *slots* e canais para cada nó. O algoritmo de agendamento, baseado em coloração de vértices, faz com que vizinhos a dois saltos de distância utilizem uma tupla (*slot*, canal) diferente.

Os autores avaliaram o protocolo utilizando simulações em MATLAB, e os resultados mostram que o protocolo permitiu o envio de pacotes através de 5 saltos com uma latência fim-a-fim inferior a 20 ms. O tamanho dos *slots* é calculado pela divisão do tamanho do pacote pela taxa de transferência máxima nominal do transceptor, e os nós são considerados perfeitamente sincronizados após a primeira transmissão, sendo dispensado o uso de períodos de guarda para superar os erros de sincronização.

Com essas suposições é possível definir um *slot* de aproximadamente 1,5 ms (no pior caso, quando são usadas mensagens de confirmação) e portanto, permitem a obtenção de latências nas simulações que dificilmente seriam obtidas implementações reais. Comparando com outros protocolos TDMA e considerando a precisão dos protocolos de sincronização, o período de 1,5 ms é demasiadamente curto para comportar os tempos de guarda necessários para evitar os erros de sincronização, o tempo para troca de canal do transceptor, e o tempo necessário para a transmissão de pacotes de controle e de dados dentro dos *slots*. O RTH-MAC não suporta mobilidade dos nós sem uma nova fase de descoberta e reconfiguração da rede.

### 2.5.2 i-MAC

O i-MAC (CHINTALAPUDI, 2010) é um protocolo TDMA multicanal voltado para máquinas de linha de produção que possuem dezenas ou até centenas de sensores acoplados.

O protocolo tem como objetivo aprender os padrões de transmissão de dados para realizar a programação e distribuição dos *slots* de tempo entre os nós. Nós que possuem alta probabilidade de gerar dados ao mesmo tempo são associados a *slots* diferentes, enquanto os que têm baixa probabilidade de gerar tráfego simultaneamente são associados ao mesmo *slot* de transmissão.

Apesar de ser altamente adaptável, o protocolo é voltado especificamente para máquinas de linha de produção com baixa mobilidade dos nós e sem variações consideráveis de topologia e tráfego. Se um nó passa a gerar uma grande quantidade de tráfego, o protocolo se reconfigura para suportar esse novo padrão. Enquanto essa reconfiguração não é finalizada podem ocorrer perdas de pacotes.

Com esse comportamento, o protocolo pode ter problemas quando submetido a redes em que os nós apresentam tráfego em rajadas, pois constantes modificações de configuração serão realizadas e perdas de pacotes ocorrerão até sua conclusão. O i-MAC suporta que o *sink* possua mais de um rádio, podendo se comunicar simultaneamente com diferentes sub-redes, mas cada sub-rede opera em apenas um canal por vez. Caso interferências externas causem perdas de pacotes, o protocolo pode ser reconfigurado para operar em outro canal.

### 2.5.3 PriorityMAC

Voltado para aplicações industriais, o PriorityMAC (SHEN et al., 2014) é um protocolo híbrido CSMA e TDMA que faz a classificação das mensagens, dando prioridade mais alta para mensagens críticas e usando métodos de acesso diferentes para nós com tráfego de maior prioridade. São definidas quatro categorias de tráfego, de acordo com sua prioridade:

- *TC1*: tráfego de alta prioridade, voltado para mensagens de emergência e segurança. Pacotes nessa classe precisam ser transmitidos com o menor atraso possível. A transmissão desses pacotes tem prioridade sobre a transmissão de pacotes das demais classes, exceto de outros pacotes classificados como TC1. O PriorityMAC depende do protocolo de roteamento para que não ocorram colisões entre transmissões simultâneas de pacotes TC1.
- *TC2*: tráfego de controle extremamente crítico. Pacotes TC2 toleram latências de dezenas de milissegundos e têm prioridade sobre pacotes nas classes TC3 e TC4. Esse tipo de tráfego pode ser bloqueado por tráfego de pacotes TC1.

- *TC3*: tráfego de controle crítico, suportando latências de 100 milissegundos a um segundo. É utilizado para tráfego frequente e menos crítico que *TC2*. O acesso para transmissão desse tipo de pacote é puramente TDMA, ao contrário de pacotes *TC1* e *TC2*.
- *TC4*: tráfego de monitoramento periódico com variações entre um segundo e uma hora. Assim como no caso de tráfego *TC3*, os pacotes *TC4* são transmitidos com método de acesso TDMA. É o tipo com menor prioridade e pode ser atrasado em função das demais classes.

O PriorityMAC permite a obtenção de latências inferiores ao protocolo MAC do padrão WirelessHART, mas ainda possui problemas quando há contenção entre pacotes com alta prioridade (categoria *TC1*), o que pode prejudicar a previsibilidade da rede justamente em situações de emergência. O PriorityMAC utiliza apenas um canal para toda a rede e o uso de multicanal nesse protocolo pode ser inviável, dado que o método de acesso para pacotes de alta prioridade (*TC1* e *TC2*) é baseado em contenção. Além disso, o protocolo não suporta mobilidade dos nós.

#### 2.5.4 Z-MAC

O Z-MAC (RHEE et al., 2008) é um protocolo híbrido que se comporta como CSMA quando há pouca contenção na rede e como TDMA quando o nível de contenção aumenta. A base de seu funcionamento é CSMA e o protocolo utiliza o escalonamento do TDMA como suporte para melhorar a solução de contenção. Quando a rede é iniciada, o protocolo realiza a alocação de *slots* e o *overhead* de comunicação dessa fase é amortizado durante o restante da operação, quando é esperado que o protocolo opere majoritariamente como CSMA.

Apesar de suportar mobilidade dos nós operando em modo CSMA, ao ocorrerem mudanças na topologia da rede é necessário executar novamente a descoberta de vizinhos, sincronização de tempo e alocação de *slots* para permitir a operação em TDMA. O Z-MAC utiliza apenas um canal de comunicação para toda a rede.

#### 2.5.5 MC-LMAC

O MC-LMAC (INCEL et al., 2011) estende o protocolo LMAC (HOESEL; HAVINGA, 2004) para usar múltiplos canais. Na configuração inicial da rede são determinados os *slots* e canais que cada nó utilizará para transmitir seus dados. A quantidade de *slots* depende da

densidade da rede, sendo determinada previamente. Os *slots* são divididos em duas partes: a primeira, chamada *CF Phase*, contém vários *subslots*, um para cada canal utilizado e a segunda parte, chamada *Split Phase*, é reservada para a transmissão de dados.

Durante a *CF Phase* os nós ouvem em um canal comum, para que o detentor do *slot* envie uma mensagem de controle indicando para qual nó ele irá transmitir pacotes durante a *Split Phase*. Na segunda parte do *slot*, o transmissor e o nó de destino se configuram para o mesmo canal e os pacotes de dados são transmitidos.

Quando ocorrem disputas, ou seja, dois transmissores tentam enviar dados ao mesmo nó, o comportamento do protocolo é imprevisível. Ambos transmissores enviam mensagens informando sua intenção de transmissão em seus respectivos *subslots*, no entanto, na fase de transmissão de dados, o nó receptor deve escolher de qual transmissor irá receber por meio de algum mecanismo de prioridade. O outro transmissor deverá aguardar seu próximo *slot* para tentar enviar seus dados.

Assim, a transmissão desses dados estará sujeita ao mecanismo usado para priorização, o que dificulta a predição de quando os dados poderão ser transmitidos. Além disso, com a divisão da parte inicial dos *slots* em um *subslot* para cada canal, é necessário aumentar o tamanho do *slot* para acomodar a seção de transmissão de dados, prejudicando a latência, ou essa seção de dados será menor, prejudicando a vazão.

### 2.5.6 TreeMAC

O TreeMAC (SONG et al., 2009) é um protocolo TDMA que organiza a rede como uma árvore e divide um ciclo em *frames*, e cada *frame* em três *slots*. Os nós calculam seu *slot* de comunicação dentro de um *frame* de acordo com a sua profundidade na árvore, e a distribuição de *frames* para cada nó é baseada na quantidade de nós em sua sub-rede. O TreeMAC segue o conceito de *fairness* (justiça), em que nós com maior demanda de acesso ao meio (por gerarem mais dados ou terem mais nós em suas sub-redes) recebem mais oportunidades de acesso. No TreeMAC a transmissão de dados é unidirecional, sendo um protocolo apenas para a coleta de dados e que usa somente um canal para toda a rede.

A configuração da rede é feita por meio de um protocolo baseado em contenção para a descoberta de vizinhança e construção de uma árvore geradora mínima. Ao obter a topologia, os *frames* são distribuídos de acordo com a necessidade de cada nó. Após a configuração da rede, cada nó tem conhecimento dos períodos em que deve se manter ativo e dos períodos que pode

desativar o transceptor. Isso reduz a necessidade de mensagens de controle, além de aumentar a previsibilidade do protocolo por evitar disputas por *slots* de transmissão.

O TreeMAC não suporta mobilidade dos nós sem que ocorra novamente a fase de descoberta de vizinhos. Outro problema é o fato de não permitir comunicação bidirecional, sendo viável apenas para aplicações com coleta de dados e transmissão para o *sink*.

### 2.5.7 DynMAC

O DynMAC (CORREIA et al., 2015) é um protocolo TDMA baseado no GinMAC (SURIYACHAI; BROWN; ROEDIG, 2010; O'DONOVAN et al., 2013) voltado para monitoramento e controle industrial. No DynMAC a rede é organizada como uma árvore e tem uma quantidade fixa de *slots* em seu *superframe*. Os *slots* são divididos em *upstream*, utilizados para transmissão de dados na direção do *sink*, para dados de monitoramento; *downstream* para a transmissão de dados do *sink* para a rede, como pacotes de controle; *adicionais* para aumentar a confiabilidade e retransmitir pacotes perdidos; *vazios* que permitem que os nós desliguem seus rádios para economia de energia.

Por ser voltado a aplicações industriais críticas, a evolução trazida ao GinMAC pelo DynMAC é em relação à resiliência da rede. Utilizando mecanismos de rádios cognitivos, o DynMAC reconfigura a RSSF para operar no melhor canal em um determinado instante. O protocolo se baseia na avaliação dos valores de RSSI (*Received Signal Strength Indicator*) e CCA (*Clear Channel Assessment*) obtidos do transceptor.

Com esses valores o protocolo constrói uma tabela e calcula um custo para cada canal em que quanto menor o custo, melhor será a qualidade da conexão. O DynMAC depende de planejamento prévio da rede e mudanças de topologia não são gerenciadas automaticamente, dependendo de uma nova configuração e inviabilizando o suporte a mobilidade. Apesar de ser capaz de mudar o canal de operação da rede quando este é degradado por interferências, a rede toda opera em apenas um canal por vez.

### 2.5.8 MCAS-MAC

O MCAS-MAC (LIM; JANG; SICHITIU, 2015) é um protocolo escalonado assíncrono e multicanal, baseado no AS-MAC (JANG; LIM; SICHITIU, 2008). Os nós enviam pacotes de controle em períodos predeterminados. Um nó tentando entrar na rede deve ouvir durante todo o intervalo de transmissão desses pacotes, uma vez em cada canal utilizado, a fim de coletar as

informações de canal e escalonamento dos vizinhos. Após essa coleta o novo nó irá conhecer em que momento e em qual canal cada vizinho opera.

Os nós determinam seu momento de operação como o ponto médio do maior intervalo de operação de seus vizinhos, e o canal selecionado é aquele que for menos utilizado na vizinhança. Após essa definição, o nó recém associado à rede faz uma escuta ociosa relativamente longa (64 milissegundos na avaliação dos autores) para garantir que nenhum outro nó selecionou o mesmo escalonamento.

Quando há um pacote para a transmissão, os nós buscam em suas informações locais o instante em que o nó de destino estará ativo e seu canal para recepção de pacotes. Antes de transmitir o nó ouve o canal para garantir que nenhuma outra transmissão esteja ocorrendo e então envia os dados em *buffer* para o destinatário.

O MCAS-MAC suporta mobilidade de nós, no entanto, o processo demanda tempo, pois escuta em vários canais por longos períodos e ainda um período adicional de escuta para garantir que o nó selecionou um escalonamento único. Essas características podem elevar o tempo de *handoff* e causar perdas de pacotes. Outro problema do protocolo é em relação ao escalonamento assíncrono, que torna a rede menos previsível quando submetida a tráfego intenso.

### **2.5.9 Mobilidade**

Muitos dos protocolos MAC para redes críticas analisados são capazes de se configurar automaticamente após os nós serem dispostos ou ocorrerem mudanças de topologia. Essa capacidade, contudo, não significa que o protocolo suporta mobilidade dos nós. Algumas soluções dependem de períodos de configuração com intensa troca de mensagens que interrompem o funcionamento da rede até que a configuração esteja completa.

A mobilidade de nós em RSSF pode ser classificada de duas formas: fraca mobilidade ou forte mobilidade (ALI; SULEMAN; UZMI, 2005). Fraca mobilidade ocorre quando as variações de topologia são raras, como aquelas devido a nós que deixam de funcionar por falhas de *hardware* ou quando novos nós são adicionados à rede. Na forte mobilidade as mudanças na topologia são mais frequentes, envolvendo movimentação física dos nós (ALI; SULEMAN; UZMI, 2005).

### 2.5.10 MA-MAC

O *Mobility-Aware Medium Access Control Protocol* (MA-MAC) (ZHIYONG; DARGIE, 2010) é uma extensão do X-MAC, usando preâmbulos curtos e operando em contenção quando a rede se mantém estática. Se o protocolo detecta alguma deterioração em uma conexão, é iniciado um processo de *handoff*, transferindo a conexão para um outro nó com o qual o sinal esteja com melhor qualidade e o restante dos dados é transmitido por meio dessa nova conexão.

O processo de detecção de mobilidade depende da estimativa da qualidade da conexão e assume que essa variação seja gradual, o que nem sempre ocorre em casos de redes móveis. Por ser um protocolo baseado em contenção, não é indicado para redes densas e com tráfego intenso.

### 2.5.11 M-TDMA

Baseado em divisão de tempo, o M-TDMA (JHUMKA; KULKARNI, 2007) é um protocolo TDMA adaptado para suportar as mudanças de topologia devido à movimentação de nós pela rede. O protocolo particiona a rede em *clusters* e cada nó pertencente a um *cluster* recebe um único *slot*. Alguns *slots* são reutilizados em diferentes *clusters* e outros *slots* são mantidos livres para futura alocação.

O M-TDMA mantém *slots* vazios em cada *cluster* para suportar mobilidade lenta dos nós. A entrada ou saída de um nó em um *cluster* só é configurada no ciclo seguinte à sua detecção no novo *cluster* e o nó só passa a operar novamente após sua configuração nesse *cluster*. O M-TDMA usa apenas um canal para toda a rede.

### 2.5.12 Sumário dos trabalhos analisados

A Tabela 2.1 apresenta um sumário dos trabalhos discutidos nesta seção. Os trabalhos selecionados apresentam soluções para o problema abordado neste trabalho, ou apresentam técnicas relevantes para a discussão e implementação do protocolo Gen3.

Na primeira coluna é indicado o nome do protocolo ou solução proposta. A segunda coluna traz o tipo de alocação empregada, sendo TDMA, CBP ou híbrida. Na terceira coluna é indicado se o protocolo tem capacidade de operação em múltiplos canais simultaneamente. Protocolos em que todos os nós utilizam o mesmo canal, mas que permitem a troca desse canal de forma global não foram considerados multicanais. A quarta coluna indica se o protocolo



suporta ou não mobilidade dos nós, e por fim, a última coluna traz de forma sucinta o foco principal do protocolo.

Tabela 2.1 – Sumarização dos trabalhos analisados.

Artigo/Protocolo	Alocação	Multicanal	Mobilidade	Foco Principal
RTH-MAC	TDMA	Sim	Não suporta	Redução de latência
i-MAC	TDMA	Não	Não suporta	Adaptação a padrões de tráfego
PriorityMAC	Híbrido	Não	Não suporta	Priorização de tipos de tráfego
Z-MAC	Híbrido	Não	Suporta	Economia de energia e adaptabilidade
MC-LMAC	TDMA	Sim	Suporta	Maximizar transmissões simultâneas
TreeMAC	TDMA	Não	Não suporta	Coleta de dados, <i>fairness</i> e baixa latência
DynMAC	TDMA	Não	Não suporta	Redução de latência, confiabilidade e resiliência da rede
MCAS-MAC	TDMA	Sim	Suporta	Economia de energia
MA-MAC	CBP	Não	Suporta	Suporte a forte mobilidade
M-TDMA	TDMA	Não	Suporta	Suporte a mobilidade (fraca)

Fonte: do autor (2017).

## 2.6 Considerações finais

As RSSF podem ser utilizadas para uma vasta gama de aplicações. Os protocolos e as técnicas utilizadas nessas redes variam com a aplicação e devem ser consideradas nos projetos. Essas redes têm se mostrado promissoras para atividades de monitoramento e controle de atividades críticas, como é o caso de tarefas em uma planta industrial. Para isso, há a necessidade de desenvolvimento de protocolos robustos e ao mesmo tempo flexíveis para essas redes.

Os protocolos de controle de acesso ao meio são determinantes para a performance geral de uma rede. A dinamicidade encontrada nas RSSF impõe desafios ainda em aberto para esses protocolos, sendo eles objeto de constante estudo e desenvolvimento. O tipo de acesso utilizado por um protocolo pode determinar o quão adequado ele é para uma determinada aplicação. Na literatura encontra-se uma tendência maior à utilização de protocolos TDMA ou híbridos para aplicações críticas de monitoramento e controle, enquanto protocolos CBP se mostram mais compatíveis com aplicações em que o foco é em economia de energia e mobilidade.

Protocolos TDMA apresentam desafios relacionados à sincronização de informações, configuração e manutenção do escalonamento. Ao introduzir a possibilidade de comunicação simultânea em múltiplos canais, novos desafios são incorporados. Apesar disso, a combinação

dessas técnicas apresenta um potencial considerável de melhoria da performance dos protocolos. Ademais, mecanismos de multiplexação estatística pouco explorados em RSSF, mas há muito aplicados em redes de telefonia, podem ser utilizados para aprimorar ainda mais esses protocolos.

Nos trabalhos discutidos nessa seção, nota-se uma maior dificuldade dos protocolos TDMA de se adaptarem às diferentes topologias de uma RSSF. Em alguns casos é necessário que se tenha um conhecimento prévio das características da rede para a definição de parâmetros do protocolo, e em outros casos pode ser necessária até a interrupção da operação dessa rede para que ela se reconfigure. Desse modo, o suporte às variações de topologia e à mobilidade é prejudicado. No entanto, durante sua operação os protocolos TDMA se mostram mais previsíveis que os protocolos CBP ou híbridos, evitando períodos de contenção que podem inviabilizar a execução de aplicações críticas.

Nenhum dos protocolos TDMA analisados se mostrou eficaz em suportar mobilidade e em mitigar os problemas de alocação de canal em situações de tráfego desbalanceado. O protocolo desenvolvido neste trabalho, que combina autoconfiguração adaptável à quantidade de nós e à topologia da rede, suporte a mobilidade e melhoria da alocação de recursos por multiplexação estatística é apresentado no próximo capítulo.

### 3 DESCRIÇÃO DO PROTOCOLO GEN3

Este trabalho apresenta o Gen3, um protocolo TDMA generalista que organiza a rede logicamente como uma árvore. O protocolo é autoconfigurável e ajusta suas configurações para suportar a entrada de novos nós, independente da densidade da rede e da distribuição destes nós pelo espaço. Cada nó associado transmite pacotes de associação para fornecer conectividade a novos nós entrando em sua área de cobertura. O Gen3 ainda utiliza múltiplos canais de comunicação para oferecer maior capacidade de vazão, redução da latência e aumento da taxa de entrega de pacotes, bem como um mecanismo de multiplexação estatística para permitir o uso oportunístico dos *slots* subutilizados.

Seja  $N = \{n_0, n_1, n_2, \dots, n_x\}$  o conjunto de nós da rede, sendo o *sink* um nó predefinido tal que  $n_{sink} \in N$ , a topologia da rede é organizada como uma árvore, tendo  $n_{sink}$  como raiz. O conjunto  $K_{n_x}$  contém os nós filhos do nó  $n_x$ , que são aqueles associados diretamente a este nó, enquanto o conjunto  $G_{n_x} \supset K_{n_x}$  contém todos os nós na sub-rede de  $n_x$  (filhos dos filhos e assim por diante). O nó pai de  $n_x$  é denotado por  $p_{n_x} \in P_{n_x}$ , sendo  $P_{n_x}$  um conjunto de no máximo um elemento ( $|P_{n_x}| \leq 1$ ), pois todo nó possui apenas um nó pai, exceto pelo *sink*, para o qual  $P_{sink} = \emptyset$ .

A profundidade de um nó na árvore, que é a distância em saltos para o  $n_{sink}$ , é dada por  $D_{n_x}$  de modo que  $D_{K_{n_x}} = D_{n_x} + 1$  e  $D_{n_x} < D_{G_{n_x}}$ , isto é, a profundidade dos nós filhos de  $n_x$  é igual à profundidade de  $n_x$  mais um e a profundidade dos nós na sub-rede de  $n_x$  é sempre maior que a profundidade de  $n_x$ . De forma análoga,  $H_{n_x}$  representa a altura da árvore a partir de  $n_x$ , dada pela distância em saltos de  $n_x$  até seu filho mais distante (o filho com maior profundidade).

Por ser um protocolo TDMA, são alocadas fatias de tempo para a operação dos nós. Essas fatias de tempo são chamadas de *frames*, representadas pelo conjunto  $F = \{f_0, f_1, f_2, \dots, f_x\}$ . O *sink* é o único nó que opera em todos os *frames*, sendo assim,  $F_{sink} = F$  e os demais nós recebem subconjuntos de  $F_{sink}$ . O método de alocação de *frames* garante que filhos de um mesmo pai operem em tempos distintos e é descrita em detalhes na Seção 3.1 (**Alocação de Frames e Duração de Ciclos**).

A Tabela 3.1 sumariza a notação apresentada, bem como outras que serão utilizadas para a descrição do protocolo no decorrer do texto.

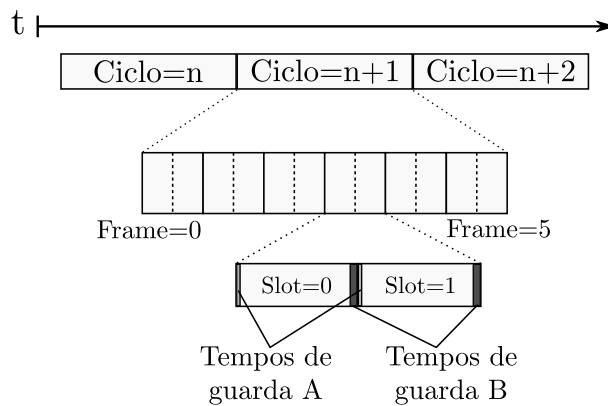
A divisão de tempo utilizada é ilustrada na Figura 3.1. O conjunto de todos os *frames* forma um ciclo, que se repete ao longo do tempo. Cada *frame* é subdividido em dois *slots*, que são compostos por dois tempos de guarda (A e B) e um período livre para comunicação.

Tabela 3.1 – Notação utilizada para a descrição do protocolo Gen3.

Notação	Descrição
$N = \{n_0, n_1, n_2, \dots, n_x\}$	Conjunto de nós da rede.
$G_x \subset N$	Conjunto de todos os nós pertencentes à sub-rede de $x$ .
$K_x \subset G_x$	Nós filhos associados ao nó $x$ .
$p_x \in P_x$	Nó pai de $x$ .
$D_x$	Profundidade de $x$ na árvore (saltos até o <i>sink</i> ).
$H_x$	Altura da árvore a partir de $x$ (saltos até o filho mais distante).
$F = \{f_0, f_1, f_2, \dots, f_x\}$	Conjunto de <i>frames</i> na rede.
$F_x \subset F$	<i>Frames</i> alocados ao nó $x$ .
$C = \{c_0, c_1, c_2, \dots, c_x\}$	Canais suportados pelo rádio.
$L \subset C$	Lista de canais utilizados pelo protocolo
$s_x^{tx} \in S = \{0, 1\}$	<i>Slot</i> em que o nó $x$ inicia a transmissão.
$s_x^{rx} \in S = \{0, 1\}$	<i>Slot</i> em que o nó $x$ aguarda por transmissões.
$T_s$	Duração de um <i>slot</i> , em segundos.
$T_c$	Duração de um ciclo, em segundos.

Fonte: do autor (2017).

Figura 3.1 – Formato de divisão de tempo do Gen3.



Fonte: do autor (2017).

No início do *slot*, o tempo de guarda A tem como objetivo contornar o erro de sincronização, garantindo que as configurações (*frame*, *slot*, *canal*) dos nós que irão se comunicar estejam corretas. A duração do tempo A varia de acordo com a precisão do protocolo de sincronização, tal que usando protocolos mais precisos pode-se reduzir esse tempo e aumentar o período livre para a comunicação.

O tempo de guarda B marca o fim do período de transmissão de dados e depende do tempo necessário para a transmissão de dois pacotes de controle e seus pacotes de confirmação. Caso a transmissão de dados do nó filho para o nó pai ocupe todo o período reservado para comunicação, ao entrar no tempo de guarda B o filho envia um pacote de controle indicando o fim da transmissão de dados para o pai, que responde com um pacote de confirmação, envi-

ando um pacote de controle para o filho contendo dados de configuração e aguardando a sua confirmação. Os pacotes de controle são detalhados na Seção 3.4 (**Pacote de Controle**).

O tempo de duração do *slot* é determinante para o desempenho da rede e tem *tradeoffs* (compromissos) que devem ser considerados. Com um *slot* muito curto, os tempos de guarda podem ocupar parte significativa desse *slot*, restando pouco tempo útil para a transmissão de dados e prejudicando a vazão dessa rede. No entanto, o *slot* de menor duração pode permitir latências menores, dado que a duração do ciclo também será menor.

Por outro lado, com um *slot* de maior duração os nós possuem um tempo maior para a transmissão de dados, atenuando também o impacto dos tempos de guarda. Porém, isso aumenta a duração do ciclo e os nós precisam aguardar mais tempo para terem uma nova oportunidade de transmissão, aumentando a latência na entrega de pacotes. Outra questão é o maior desperdício de tempo gerado quando o tráfego da rede é baixo ou desbalanceado, pois as transmissões de dados não ocuparão todo o período de transmissão do *slot*. O restante desse *slot* poderia ser alocado a outros nós, como será mostrado com o uso da multiplexação estatística no Gen3, mas ainda assim há casos em que ela não pode ser empregada, e o tempo subutilizado do *slot* terá impacto negativo no desempenho da rede.

Por meio de testes empíricos e de avaliações preliminares do Gen3, a duração do *slot* foi definida em 20 milissegundos. O tempo de guarda A foi definido com duração de 1 milissegundo, com base na análise dos protocolos de sincronização discutidos anteriormente, e o tempo de guarda B com 2 milissegundos, tempo suficiente para a transmissão dos pacotes de controle do protocolo, que serão discutidos no decorrer deste capítulo.

### 3.1 Alocação de *frames* e duração de ciclos

A quantidade de *frames* em que um ciclo é dividido depende da quantidade de nós associados à rede. Em casos onde ocorrem variações de topologia e com nós que se movimentam, este valor pode oscilar até que ocorra a reorganização. Portanto, em uma rede considerada estável, a quantidade de *frames* é igual à quantidade de nós ( $|F| = |N|$ ).

O *sink* é o único nó que opera em todos os *frames*, logo,  $F_{sink} = F$ . Os *frames* de operação dos filhos do nó *sink* são um subconjunto dos *frames* de operação do *sink*, como descrito na relação da Equação 3.1. Generalizando esse conceito, a Equação 3.2 determina que o conjunto de *frames* de um nó  $n_y$ , filho de um nó  $n_x$ , seja sempre um subconjunto dos *frames* de  $n_x$ . Isso é válido para todos os nós da rede com algum filho associado. Além disso, a Equação

3.3 garante que os *frames* alocados por  $n_x$  a um filho  $n_y$  sejam disjuntos dos *frames* alocados a um filho  $n_{y+1}$ , de modo que não ocorram colisões entre eles. A Figura 3.2 ilustra essa alocação.

A partir das definições em 3.2 e 3.3 também é possível garantir a relação da Equação 3.4, que diz que nenhum nó na sub-rede de  $n_y$  será escalonado ao mesmo *frame* que qualquer outro nó na sub-rede de  $n_{y+1}$ .

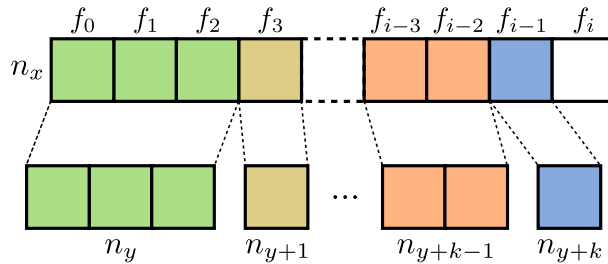
$$F_{K_{sink}} \subset F_{sink} \quad (3.1)$$

$$\forall n_y \in K_{n_x}, F_{n_y} \subset F_{n_x} \quad (3.2)$$

$$\forall n_y \in K_{n_x}, F_{n_y} \cap F_{n_{y+1}} = \emptyset \quad (3.3)$$

$$\forall n_z \in G_{n_y}, \forall n_k \in G_{n_{y+1}}, F_{n_z} \cap F_{n_k} = \emptyset \quad (3.4)$$

Figura 3.2 – Alocação de *frames* de um nó pai para seus nós filhos.



Fonte: do autor (2017).

A quantidade de *frames* de um nó  $x$  é dada por  $|F_x| = |G_x| + 1$ , isto é, os nós possuem uma quantidade de *frames* igual à quantidade de nós em sua sub-rede mais um. Essa forma de distribuição visa dar mais oportunidades de transmissão a nós que necessitam encaminhar pacotes de mais nós filhos, seguindo o conceito de alocação justa de tempo (*fairness*) proposto em Song et al. (2009). O *frame* adicional não possui nó filho associado e é utilizado pelo nó para a transmissão de pacotes de descoberta, para permitir que outros nós se associem à rede.

O novo filho trabalhará no *frame* em que se associou e o nó pai irá requisitar um *frame* adicional para continuar a transmissão dos pacotes de descoberta. Quando algum nó filho se desconecta, o nó pai devolve os *frames* em excesso para que volte a ficar com apenas um adicional. A requisição e a devolução de *frames* é detalhada na Seção 3.2.

A quantidade de *frames* varia de acordo com a quantidade de nós e com as mudanças de topologia, tornando a duração dos ciclos variável. A duração, em segundos, de um ciclo é dada pela Equação 3.5, sendo duas vezes a duração de um *slot* (pois cada *frame* é composto por dois *slots*), multiplicado pela quantidade de *frames* que o *sink* possui. Apenas o *sink* pode adicionar e remover *frames* da rede, alterando o tamanho do ciclo. Quando isso ocorre, uma

nova configuração é propagada pela rede para que os nós se reconfigurem com o novo tamanho de ciclo.

$$T_c = 2T_s \times |F_{sink}| \quad (3.5)$$

Ao gerar uma nova configuração, o *sink* faz uma estimativa de quanto tempo levará para que ela seja transmitida a todos os nós. A estimativa retorna a quantidade aproximada de ciclos necessários para que a configuração seja propagada e é utilizada para determinar o ciclo de *deadline*, que é transmitido junto com as novas configurações. O *sink* estima a altura da árvore a partir das informações recebidas nos pacotes de controle dos filhos. O ciclo de *deadline* é calculado somando-se o valor do ciclo atual (em que a nova configuração está sendo gerada) com a estimativa de altura da rede.

Os nós recebem as configurações da rede nos pacotes de controle do nó pai. Dentre essas informações, quatro determinam se o nó está recebendo uma configuração atualizada: *frame* inferior, quantidade de *frames* do nó, quantidade total de *frames* da rede e ciclo de *deadline*. Se o valor recebido de qualquer um desses valores for diferente do que o nó tem armazenado localmente, significa que a configuração recebida é nova. Essa configuração é armazenada em uma estrutura separada e o nó continua utilizando a configuração antiga até o ciclo de *deadline*, quando a configuração antiga é descartada e o nó passa a utilizar a configuração atualizada da rede.

### 3.2 Requisição e devolução de *frames*

A requisição e a devolução de *frames* é realizada por meio de requisições dos nós filhos enviadas nos pacotes de controle. Para isso, é utilizado o campo *Frame Request* do pacote de controle, que será descrito na Seção 3.4 (**Pacote de Controle**). Um valor positivo nesse campo indica que o nó necessita de mais *frames*, enquanto um valor negativo indica que o nó precisa ter sua quantidade de *frames* reduzida. Enquanto o nó não recebe uma configuração com a quantidade atualizada de *frames*, ele continua fazendo a requisição ao nó pai em todo pacote de controle transmitido. Desse modo, o nó pai pode identificar quando a quantidade de *frames* requisitada pelo nó filho foi modificada, além de não precisar de mecanismos para identificar requisições duplicadas.

Caso a quantidade de nós seja maior que a quantidade atual de *frames*, as requisições para adição de *frames* são repassadas até chegarem ao *sink*, que adiciona a quantidade necessária de *frames* ao ciclo, gera uma nova configuração e a transmite pela rede. Enquanto a rede não

chega ao ciclo de *deadline* o *sink* não gera novas configurações, evitando que ocorram conflitos de configurações.

Quando algum nó filho se desconecta, seu pai passa a enviar pacotes de descoberta nos *frames* vagos deixados enquanto requisita a devolução dos *frames* em excesso, de forma que volte a ficar com apenas um *frame* adicional. Quando uma nova configuração for propagada à rede, o nó superior verifica se a requisição de devolução de *frames* ainda está pendente e repassa a nova configuração ao filho removendo dele a quantidade de *frames* solicitada. Com isso, os *frames* vagos passam ao nó superior e três ações podem ser tomadas: (i) distribuição desses *frames* a outros nós filhos quando uma nova configuração vinda do *sink* for recebida; (ii) associação de novos filhos, pois ocorre a transmissão de pacotes de descoberta nos *frames* vagos; (iii) solicitação de devolução desses *frames* a seu nó pai, que tomará as mesmas ações até que a requisição chegue ao *sink* e este remova os *frames* em excesso do ciclo.

Como a remoção de *frames* de um nó filho só é realizada quando o nó pai recebe uma nova configuração geral da rede, é necessário que uma configuração seja gerada pelo *sink* periodicamente. Caso contrário, a requisição para remoção de *frames* poderia ficar retida em um nó, sem nunca chegar ao *sink* e permitir a remoção desses *frames* do ciclo. Nesse caso, o ciclo teria uma quantidade de *frames* maior que o necessário, impactando no desempenho geral da rede. A atualização periódica é gerada caso o último ciclo de *deadline* tenha ocorrido a mais de  $2 \times H_{sink}$  ciclos.

Um parâmetro denominado *Settle Cycles* (ciclos de estabilização) determina como o protocolo reage à associação de novos nós e define quantos ciclos devem ser aguardados após a associação de um filho para que seja feita a requisição de um *frame* adicional. Caso seja o *sink*, o nó aguarda esse tempo para gerar uma nova configuração da rede. O objetivo desse parâmetro é evitar a mudança constante de configurações em caso de nós que se associam com conexões instáveis, ou nós com alta mobilidade que constantemente se conectam e desconectam em diversos pontos da rede.

Outro parâmetro define quantos ciclos um nó recém associado deve aguardar antes de iniciar a transmissão de pacotes de descoberta para permitir a associação de outros nós. Esse parâmetro, denominado *CyclesToTxAssociation*, não é passado diretamente ao protocolo, mas calculado a partir do parâmetro *Settle Cycles* e da quantidade de saltos do nó para o *sink* por meio da Equação 3.6.



$$CyclesToTxAssociation = (2 \times D) + SettleCycles \quad (3.6)$$

O objetivo do parâmetro *CyclesToTxAssociation* é fazer com que um nó só passe a fornecer associação a nós filhos se tiver uma conexão estável à rede. Isso impede que nós se associem a outros nós com conexões instáveis ou transientes, como aqueles que se movimentam por uma região. A intenção de que este parâmetro seja baseado no *Settle Cycles*, e maior que ele, é que os nós pais tenham chance de buscar uma nova configuração com mais *frames* antes que os filhos comecem a permitir a associação de outros nós.

### 3.3 Configuração de *slots* e canais

Em profundidade os nós compartilham *frames* e dois mecanismos são utilizados para evitar a ocorrência de colisões. O primeiro mecanismo é o uso alternado de *slots*, determinado pela profundidade do nó na árvore. Os nós calculam em qual dos dois *slots* devem iniciar as transmissões para o nó pai (Equação 3.7) e em qual dos *slots* devem aguardar por transmissões de um nó filho associado (Equação 3.8).

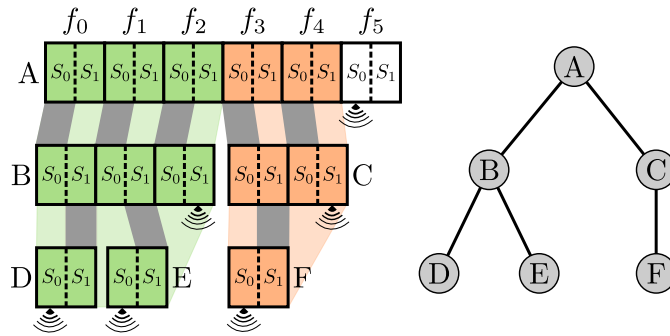
$$S_x^{tx} \in \{0, 1\} = (D_x - 1) \text{ mod } 2 \quad (3.7)$$

$$S_x^{rx} \in \{0, 1\} = D_x \text{ mod } 2 \quad (3.8)$$

A Figura 3.3 ilustra como a combinação de alocação disjunta de *frames* e uso alternado de *slots* garante que as transmissões ocorram em tempos diferentes em uma rede. Os *frames* do nó A são alocados aos filhos B e C de modo que recebam conjuntos separados. Dessa forma, os nós na sub-rede de B (nós D e E) e na sub-rede de C (nó F) sempre irão operar em tempos distintos. No entanto, a figura mostra que os nós A, B e D compartilham o *frame* 0, e para evitar colisões neste *frame*, a comunicação entre A e B ocorre no *slot* 0 ( $S_0$ ), enquanto a comunicação entre B e D ocorre no *slot* 1 ( $S_1$ ). O mesmo ocorre entre os nós A, B e E, e entre A C e F. No último *frame* de cada nó, no *slot* em que ele se comunicaria com um filho, são transmitidos os pacotes de descoberta.

O segundo mecanismo para evitar colisões é o uso de múltiplos canais, também determinado pela profundidade do nó na árvore. Vários nós em uma mesma sub-rede podem operar no mesmo *slot* ao mesmo tempo, porém em profundidades e canais diferentes. Para determinar esses canais o Algoritmo 1 gera uma lista de canais de acordo com a quantidade de canais dis-

Figura 3.3 – Alocação de *frames* e *slots* para comunicação livre de colisões.



Fonte: do autor (2017).

poníveis no rádio e a quantidade definida para uso pelo protocolo. Na linha 1, a função recebe como parâmetro a quantidade de canais que o protocolo MAC usará. Na linha 2 é lida a quantidade de canais suportados pelo rádio e na linha 3 a lista é inicializada. As linhas 4 e 5 definem a base (o primeiro canal a ser usado na lista) e o *step*, que dará o espaçamento de canais da lista. Nas linhas 6 a 9 um laço preenche a lista de acordo com a quantidade de canais utilizados e na linha 10 a função retorna a lista gerada.

---

**Algoritmo 1** Criação da lista de canais

---

```

1: procedure SETUPCHANNELLIST(channels)
2:    $MAX\_CHANNELS \leftarrow |ChSupported|$  ▷ Channels supported by radio
3:    $list \leftarrow null$  ▷ Initialize channel list
4:    $base \leftarrow 0$ 
5:    $step \leftarrow MAX\_CHANNELS / (channels - 1)$ 
6:   for ( $i = 0$ ;  $i < channels$ ;  $i++$ ) do
7:      $list[i] \leftarrow base$ 
8:      $base \leftarrow base + step$ 
9:   end for
10:  return  $list$  ▷ List of channels to be used
11: end procedure

```

---

Esse mecanismo de lista permite mapear canais reais do rádio tal que sejam usados canais mais distantes, a fim de evitar interferências. Além disso, permite que os nós negociem localmente a substituição de canais da lista com alto nível de ruído por canais com menor ruído ou livres de interferências. No entanto, a troca de um determinado canal definido pelo Algoritmo 1 deve ser feita dentro de um limite. Seja  $C$  o conjunto de canais suportados pelo rádio,  $L$  os canais selecionados para uso e  $i$  o índice de um elemento de  $L$ , o canal referenciado por  $i$  só pode ser alterado por um outro dentro do limite  $\left[ i \times \frac{|C|}{|L|}, (i+1) \times \frac{|C|}{|L|} \right]$ .

Em uma rede em que o rádio suporte 120 canais e o MAC esteja configurado para usar 4 canais, a lista gerada será  $L = [0, 40, 80, 119]$  e o canal referenciado pelo índice  $i = 0$  (canal 0) poderá variar entre  $[0, 30[$ , o canal referenciado por  $i = 1$  (canal 40) entre  $[30, 60[$ , e assim por diante. Para efetuar a negociação e a troca de canais da lista de forma dinâmica seriam necessários métodos de rádios cognitivos para as fases de Sensoriamento, Decisão, Compartilhamento e Mobilidade de Espectro (AKYILDIZ et al., 2008) que estão fora do escopo deste trabalho.

Após a definição da lista de canais os nós definem sua configuração de *slots*. A configuração de *slots* determina em qual canal o nó irá trabalhar em cada *slot*, qual seu *slot* para comunicação com o pai e qual *slot* para comunicação com os filhos. Essa configuração é feita somente após o nó se associar à rede e saber sua profundidade.

O Algoritmo 2 faz a configuração dos *slots*. Essa configuração é a mesma em todos os *frames*, ou seja, basta definir como o nó deve operar no primeiro e no segundo *slots* e essa configuração se repetirá em todos os *frames*. Para isso, é utilizada uma estrutura de dados que indica se no *slot* o nó deve transmitir dados ao nó pai ou aguardar por transmissões dos filhos e qual canal utilizar em cada *slot*.

---

**Algoritmo 2** Definição de configuração de *slot*.

---

```

1: procedure SETUP_SLOTS(config)
2:    $chSlot0 \leftarrow 0$ 
3:    $chSlot1 \leftarrow config.channels - 1$ 
4:    $iterations \leftarrow config.hopsFromSink \bmod (2 \times config.channels)$ 
5:   for ( $i = 0$ ;  $i < iterations$ ;  $i++$ ) do
6:     if  $i \bmod 2$  then
7:        $chSlot0 \leftarrow chSlot0 + 1$ 
8:        $chSlot0 \leftarrow chSlot0 \bmod config.channels$ 
9:     else
10:       $chSlot1 \leftarrow chSlot1 + 1$ 
11:       $chSlot1 \leftarrow chSlot1 \bmod config.channels$ 
12:    end if
13:  end for
14:   $config.nodeSlot[0].channel \leftarrow channelList[chSlot0]$ 
15:   $config.nodeSlot[1].channel \leftarrow channelList[chSlot1]$ 
16:   $config.nodeSlot[(config.hopsFromSink - 1) \bmod 2].transmit \leftarrow true$ 
17:   $config.nodeSlot[config.hopsFromSink \bmod 2].transmit \leftarrow false$ 
18: end procedure

```

---

O algoritmo recebe como parâmetro uma estrutura contendo configurações do nó, onde busca informações sobre a quantidade de canais utilizados pelo protocolo e profundidade do nó na rede. Nas linhas 2 e 3 os canais em cada *slot* são inicializados, com o *slot* 0 iniciando

com o primeiro canal da lista e o *slot* 1 com o último canal da lista. A quantidade máxima de combinações (*slot*, canal) é dada por  $2 \times |L|$ . Com base nisso, na linha 4 é calculada a quantidade de iterações necessárias para se chegar à configuração correta para o nó.

O laço das linhas 5 a 13 determina o índice do canal a ser utilizado em cada *slot* e nas linhas 14 e 15, o valor do canal é buscado na lista e armazenado na estrutura de configuração de *slots*. Na linha 16 é definido o *slot* em que o nó inicia as transmissões, enviando pacotes para seu nó pai, e na linha 17 o *slot* em que aguarda por transmissões de nós filhos, ambos com base na informação profundidade.

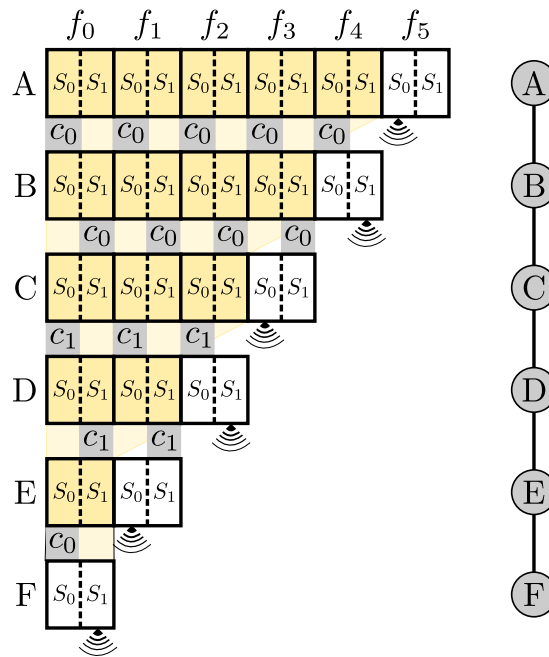
Com esses mecanismos é possível reduzir a probabilidade de interferências em profundidade alternando os *slots* de comunicação, além de transmitir em canais diferentes quando os *slots* forem compartilhados. No entanto, ainda é possível que ocorram colisões em *slots* compartilhados dependendo da quantidade de canais selecionados para uso e do valor utilizado no parâmetro *Settle Cycles*.

Com um valor baixo de *Settle Cycles* os nós aguardam menos tempo para permitir a associação de novos filhos, fazendo com que a árvore cresça em altura mais rapidamente. Dessa forma, é possível que alguns nós sejam escalonados para transmitir em um mesmo conjunto de *frame*, *slot* e canal, mesmo estando fisicamente próximos, gerando colisões e eventualmente a desconexão de nó, atrasando a estabilização da rede.

A Figura 3.4 ilustra uma rede em linha com a alocação dos *frames* entre os nós, o uso alternado de *slots* e sua combinação com multicanal para evitar colisões. No exemplo, a rede utiliza apenas 2 canais diferentes para as comunicações. A parte sombreada entre os nós indica em qual *slot* ocorre a comunicação em cada *frame* e o valor  $c_0$  ou  $c_1$  diz em qual canal da lista a comunicação ocorre.

Entre os nós A, B e C as transmissões ocorrem sempre no canal  $c_0$ . O nó C trabalha em  $c_1$  no *slot*  $S_0$  e em  $c_0$  no *slot*  $S_1$ . Enquanto C se comunica com D em  $S_0$  e nos *frames*  $f_0$  a  $f_2$ , os nós A e B também se comunicam, mas em canais diferentes. A reutilização de um mesmo conjunto de *frame*, *slot* e *canal* só ocorre a partir de E e F, e quando isso ocorre, assume-se que estejam distantes o suficiente para que não causem colisões.

Se os nós E e F estiverem muito próximos dos nós A e B, essas comunicações ocorrendo ao mesmo tempo e no mesmo canal podem causar colisões e a eventual desconexão dos nós. Esses nós irão se reconectar à rede e, devido aos parâmetros que governam a requisição e a devolução de *frames*, a nova configuração dessa rede será diferente daquela que causou colisões.

Figura 3.4 – Alocação alternada de *slots* e canais.

Fonte: do autor (2017).

Ainda assim, analisando a Figura 3.4 verifica-se que utilizando apenas 2 canais foram necessários 5 saltos para que uma configuração fosse reutilizada. Com 3 canais isso aconteceria em 7 saltos, com 4 canais ocorreria em 9 saltos e assim por diante, mostrando que com poucos canais adicionais a probabilidade de colisões é reduzida.

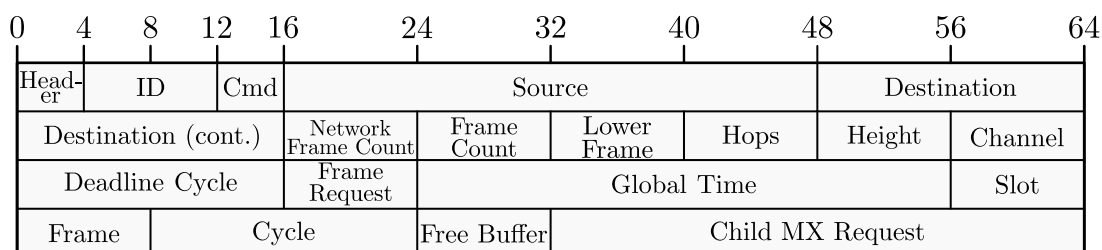
Deve-se notar que o Gen3 usa multicanal apenas em profundidade. Por ser voltado para RSSF cujos nós possuem apenas um rádio, o emprego de multicanal em largura é desnecessário, dado que a multiplexação em largura é feita no domínio do tempo. Mesmo que fossem utilizados canais diferentes em largura, o pai poderia se comunicar apenas com um nó filho por vez, pois seu único rádio seria ocupado. Usando multicanal em largura, a alocação de canais se tornaria mais complexa sem benefício real de desempenho ao protocolo.

### 3.4 Pacote de controle

Todas as ações de configuração e coordenação de comunicações se dão por meio do pacote de controle descrito na Figura 3.5. O primeiro campo é o cabeçalho (*Header*) que define o tipo do pacote. Em seguida, um campo de identificação (*ID*) é incrementado a cada novo pacote enviado, de modo que os nós possam detectar pacotes replicados. O terceiro campo contém um

comando (*Cmd*), que define como o pacote e os campos subsequentes serão processados. Os comandos estão descritos na Tabela 3.2.

Figura 3.5 – Formato do pacote de controle.



Fonte: do autor (2017).

Tabela 3.2 – Comandos do pacote de controle.

Comando	Nome	Descrição
C1	C_COMM_ASSOCIATION	Comando de descoberta para associação de novos nós.
C2	C_COMM_ASSOCIATION_REPLY	Resposta a um pacote de descoberta recebido.
C3	C_COMM_DISCONNECT	Comando para desconexão explícita de um nó.
C4	C_COMM_WAITING_DATA	Indicação de que o nó terminou de enviar seus pacotes e está pronto para receber pacotes.
C5	C_COMM_MXSTAT_REQUEST	Requisição de pacotes por meio de multiplexação estatística.

Fonte: do autor (2017).

Os próximos campos trazem os endereços de origem e destino (*Source* e *Destination* respectivamente). O campo *Network Frame Count* indica a quantidade de *frames* do ciclo, *Frame Count* é utilizado pelo nó pai para indicar quantos *frames* foram alocados ao nó filho e em *Lower Frame* o nó pai informa qual é o primeiro *frame* de operação deste filho. A partir dos campos *Frame Count* e *Lower Frame* o nó calcula em quais *frames* deve operar, podendo desativar o rádio nos demais *frames* para economizar energia.

O campo de *Hops* é utilizado para determinar a profundidade do nó na árvore, ou seja, a quantos saltos ele está do *sink*. Este valor é utilizado para determinar as configurações de *slot* de transmissão e canais de comunicação. O próximo campo, *Height*, é utilizado para determinar a altura da árvore a partir de um determinado nó, dada pela maior distância entre o nó e seu filho mais distante. A altura da árvore é utilizada quando uma nova configuração é gerada para determinar em qual ciclo a nova configuração entra em vigor. O campo *Channel* é destinado à

notificação de mudanças na quantidade de canais ou negociação de mudança de canal devido a interferências externas.

Quando uma nova configuração é recebida pelo nó, ela é armazenada e a rede continua utilizando a configuração antiga até o início do ciclo determinado no campo *Deadline Cycle*. Quando a rede chega a este ciclo, a configuração atual é descartada e a nova configuração passa a ser utilizada. Este valor é calculado somando-se a altura da árvore ao número de ciclo atual e visa garantir que todos os nós recebam a nova configuração antes que esta seja efetivada.

Por meio do campo *Frame Request* um nó pode requisitar mais *frames* a seu nó pai, ou então, no caso de desconexão de filhos, pode requisitar a redução da sua quantidade de *frames*. Os campos de tempo global (*Global Time*), *Slot*, *Frame* e ciclo (*Cycle*) são utilizados para a sincronização dos nós. Os nós são sincronizados com o relógio interno do *sink*, por meio da estimativa de tempo global propagada no campo *Global Time*. Os valores atuais da rede de *slot*, *frame* e ciclo são transmitidos nos demais campos.

Por fim, os campos de *buffer* livre (*Free Buffer*) e *Child MX Request* são usados para multiplexação estatística. O nó pai especifica no campo *Child MX Request* o endereço do filho para o qual está solicitando dados via multiplexação estatística e informa no campo *Free Buffer* quanto espaço ainda possui para armazenar os pacotes recebidos do nó filho.

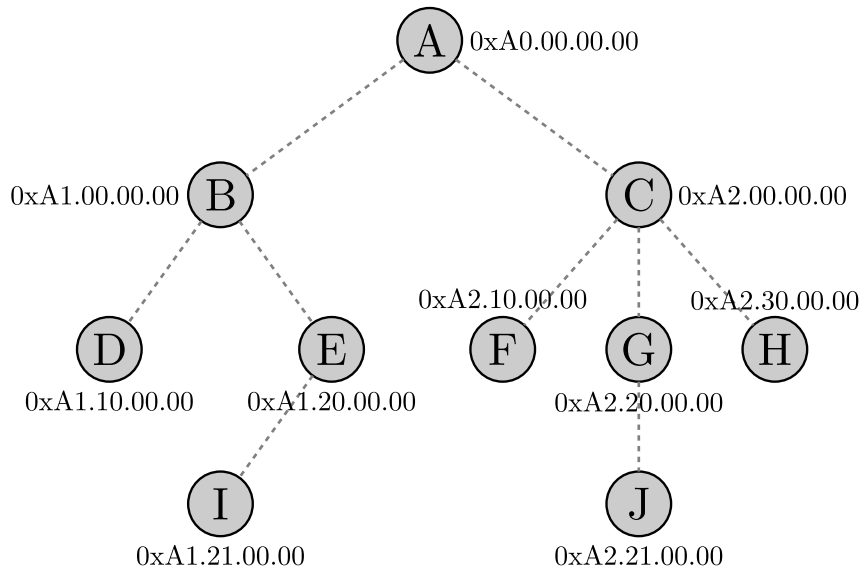
### 3.5 Endereçamento e encaminhamento

Os endereços dos nós são atribuídos dinamicamente no momento da associação, sendo que o único endereço previamente determinado é o do *sink*. O formato de endereçamento depende da quantidade de bits usados a cada salto. Neste trabalho, foram usados 4 bits por salto, logo, o endereço do *sink* ocupa os 4 bits mais significativos do campo de endereço. A cada salto são usados mais 4 bits à direita, o que permite uma rede com no máximo 7 saltos até 15 filhos por nó (para endereços de 32 bits).

Com esse formato, se o *sink* possui o endereço (em hexadecimal) 0xA0.00.00.00, seus filhos terão endereços de 0xA1.00.00.00 a 0xAF.00.00.00, respectivamente com as máscaras 0xF0.00.00.00 para o *sink* e 0xFF.00.00.00 para os filhos do *sink*. O endereçamento pode ser modificado com outros formatos de máscara para suportar divisões diferentes de endereço e aumentar a escalabilidade da rede. O formato atual foi adotado no desenvolvimento e na descrição do trabalho por ser mais simples e intuitivo.

A Figura 3.6 ilustra o endereçamento utilizado no trabalho, com o nó A como *sink* e os demais nós comuns associados formando uma árvore de altura 3.

Figura 3.6 – Formato de endereçamento utilizado no trabalho.



Devido à limitação de saltos possíveis com um tamanho de endereço limitado, quando a máscara de um nó ocupar os bits menos significativos de seu endereço o nó não transmite pacotes de descoberta, por ser impossível atribuir um endereço válido ao nó filho. Dessa forma, um nó tentando se associar só encontrará pacotes de descoberta de nós que ainda não chegaram no limite de altura da rede.

Com esse formato de endereçamento os nós são capazes de encaminhar os pacotes da rede sem a necessidade de tabelas de roteamento. Quando um novo pacote é colocado em *buffer*, o Algoritmo 3 determina para qual nó vizinho o pacote deve ser encaminhado. Para isso, o endereço de próximo salto do pacote é calculado a partir do endereço de destino final contido no pacote e do endereço do nó atual.

Na linha 1, o algoritmo recebe o endereço final de destino contido no pacote. A condicional da linha 2 testa se o endereço de destino é numericamente menor que do nó atual, o que determina que o pacote deve ser encaminhado para o nó pai, que tem o endereço armazenado na variável *gatewayAddress*.

A condicional da linha 5 determina se o pacote deve ser encaminhado ao nó pai (*gateway*) mesmo que o endereço de destino seja numericamente maior que o do nó atual. Esses casos ocorrem quando um pacote deve ser encaminhado a um outro nó em uma sub-árvore di-



ferente da atual. A não satisfação das condicionais das linhas 2 e 5 determina que o pacote deve ser encaminhado a algum nó filho, cujo endereço é calculado utilizando a operação da linha 8.

---

**Algoritmo 3** Algoritmo para determinar para qual vizinho um pacote deve ser encaminhado.

---

```

1: procedure GETNEXTHOPADDRESS(destAddress)
2:   if destAddress < myAddress then
3:     return gatewayAddress                                ▷ Send to parent node
4:   end if
5:   if (mask & destAddress) XOR myAddress then
6:     return gatewayAddress                                ▷ Send to parent node
7:   end if
8:   nextChild = ((mask >> 4) | 0xF0000000) & destAddress
9:   return nextChild                                       ▷ Send to correct child
10: end procedure

```

---

### 3.6 Associação

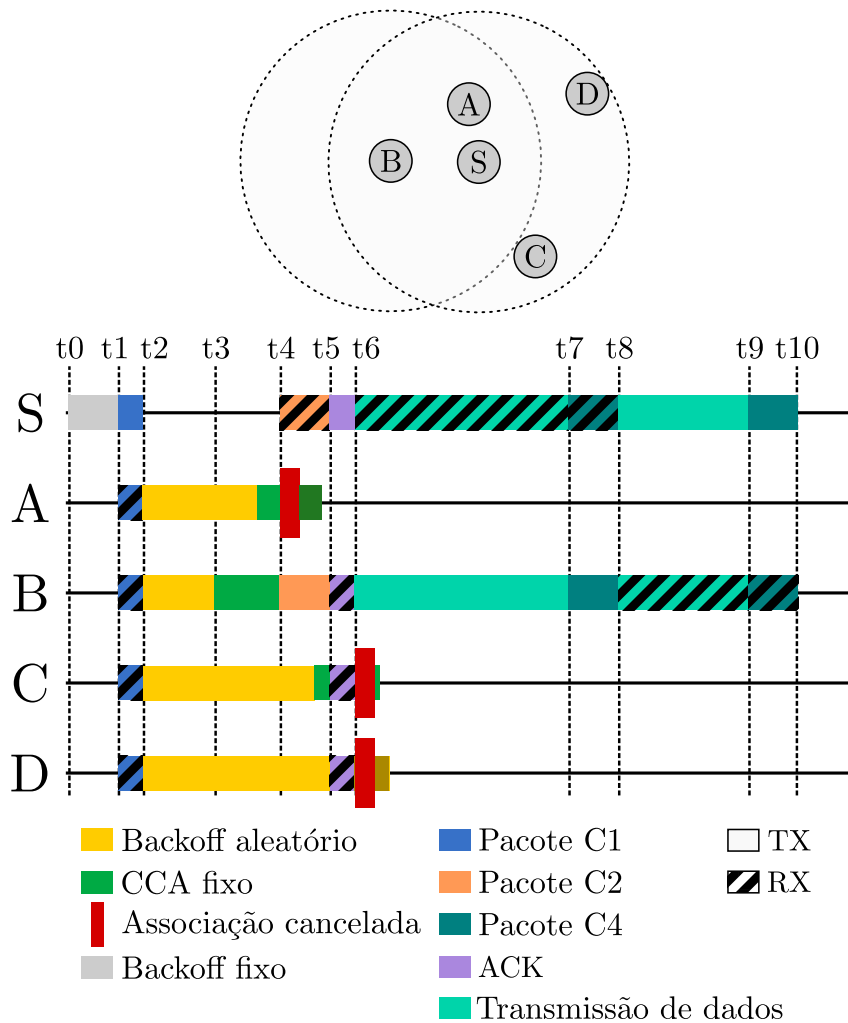
A rede é iniciada tendo apenas o *sink* associado e com um único *frame*, no qual ele transmite pacotes de controle com o comando de descoberta (pacote C1) no *slot* 0 (determinado pela Equação 3.8). Esse pacote carrega as informações de tempo global (valor do relógio interno do *sink*), valores atuais de *slot*, *frame* e ciclo da rede, quantidade total de *frames*, quantidade de *frames* que o novo nó receberá (sempre igual a um para novos nós) e a quantos saltos ele está do *sink*.

O endereço de origem é preenchido com o endereço do nó atual e o endereço de destino com o endereço de *broadcast*, dado que qualquer nó pode receber este pacote. O campo *Child MX Request* do pacote tem um tratamento diferente com este comando e contém o endereço que o novo nó assumirá na rede.

A associação de um novo nó ocorre no período de um *slot*. O pacote de controle de comando C1 é enviado no início desse *slot*, as demais trocas de mensagens e *backoffs* necessários ocupam outra parcela desse *slot*, e o tempo restante pode ser utilizado para a transmissão de dados da aplicação.

A Figura 3.7 ilustra as trocas de mensagens realizadas para a associação de um novo nó. A figura traz uma topologia com cinco nós, sendo um deles o *sink* (S) e outros quatro nós tentando entrar na rede (A, B, C e D). A segunda parte da imagem traz a linha do tempo com as trocas de mensagens dentro de um *slot*.

Figura 3.7 – Troca de mensagens para a associação de um novo nó.



Fonte: do autor (2017).

Em  $t_0$ , o nó S aguarda um tempo de guarda fixo após o início do *slot* e transmite um pacote C1 para descoberta de novos nós ( $t_1$ ). Os demais, que estão todos no alcance de S, recebem o pacote e iniciam um *backoff* aleatório em  $t_2$ , durante o qual continuam a ouvir por transmissões de outros nós. O nó B é o primeiro a encerrar o *backoff* aleatório ( $t_3$ ) e a iniciar um período fixo de checagem de canal livre (CCA - *Clear Channel Assessment*).

Ao terminar o CCA, em  $t_4$ , B transmite um pacote C2, como resposta ao pacote de associação recebido. O nó A que está no período de CCA e também no alcance de B detecta essa transmissão, cancela sua tentativa de associação e desliga seu rádio até o próximo *slot*. O *sink* recebe o pacote de B e responde com um pacote de confirmação (ACK) em  $t_5$ . Em  $t_6$  os demais nós (C e D), que não estavam no alcance de B e não detectaram sua tentativa de associação, recebem o (ACK) de S confirmando que outro nó está concluindo sua associação.

Com isso, C e D também cancelam suas tentativas de associação e podem desativar seus rádios até o próximo *slot*.

No pacote de controle C2, de resposta a um comando de associação, o campo de ID é preenchido com um número aleatório gerado pelo nó ou com algum ID único que este tenha. Esse valor é transmitido pelo nó pai no ACK e o nó se associando compara o valor recebido com o que enviou. Caso os valores sejam diferentes o nó cancela sua associação, pois há a possibilidade ter ocorrido um conflito caso os *backoffs* não tenham sido aleatórios o suficiente.

Caso sejam usados números aleatórios no campo de ID e dois nós gerarem o mesmo número, os dois se associarão ao mesmo pai no mesmo *frame* e no mesmo *slot*. Durante a operação da rede esse conflito eventualmente será identificado, seja por colisões na transmissão de pacotes ou pela duplicidade de pacotes de controle, e o nó pai irá determinar a desassociação desses filhos por meio de um pacote de controle C3. Esse é um fenômeno raro, mas que pode ocorrer durante a formação da rede ou com a mobilidade de vários nós, e é tratado pelo Gen3 sem impactos relevantes em sua operação.

De  $t6$  em diante a comunicação entre B e S ocorre normalmente. Primeiro B envia os pacotes destinados a S que possui em *buffer* e ao fim das transmissões envia um pacote de controle C4 ( $t7$ ), informando que não tem mais dados a transmitir. O nó S, por sua vez, transmite os dados que tenha em *buffer* destinados a B ( $t8$ ) e finaliza também com um pacote C4 ( $t9$ ). A partir de  $t10$  ambos podem desativar seus rádios até o próximo *slot*.

Ao final desse processo o nó B estará associado ao *sink* (S), e após o período definido por *CyclesToTxAssociation* irá começar a transmitir pacotes para permitir a associação de novos nós, que segue o mesmo procedimento apresentado na Figura 3.7.

A entrada de novos nós na rede pode se dar com ou sem um canal de controle comum (CCC) dependendo da configuração do protocolo. Se configurado para usar CCC, as mensagens de descoberta são transmitidas sempre na frequência mais alta suportada pelo rádio. Os nós tentando se associar ficam ouvindo continuamente neste canal até se associarem em algum ponto da rede. Essa configuração permite uma associação (e também reassociação) mais rápida, dado que os nós não precisam varrer vários canais tentando encontrar algum nó associado transmitindo pacotes de descoberta em um dado momento. No entanto, usar um CCC deixa a rede mais vulnerável a ataques ou interferências externas, pois a entrada de novos nós ou reconexão de nós depende da qualidade de um único canal.

Na operação sem CCC os nós devem conhecer previamente quantos canais serão utilizados para que possam gerar a lista de canais. A transmissão dos pacotes de descoberta é realizada no canal em que o nó opera, de acordo com a configuração determinada no Algoritmo 2.

Os nós tentando entrar na rede utilizam um mecanismo incremental de escuta nos canais. Primeiro, ouvem durante um *frame* em cada canal da lista e ao chegar no último canal, passam a ouvir dois *frames* em cada canal. O tempo de escuta em cada canal é incrementado até um valor predeterminado de 100 *frames*, quando o nó volta a ouvir um *frame* em cada canal, repetindo todo o processo. Com esse formato, um nó tentando entrar na rede ouve cada vez mais tempo em um mesmo canal em uma determinada região, aguardando pelas transmissões de algum nó já associado.

Se a escuta em várias frequências fosse demasiadamente curta, poderia ocorrer o caso de nunca coincidir o canal de escuta com o canal de transmissão dos pacotes de associação. O mesmo poderia ocorrer com um tempo de escuta muito longo em um mesmo canal, por isso, o Gen3 utiliza esse esquema variável e incremental de escuta por pacotes de associação.

O mecanismo sem CCC pode ajudar a deixar a rede mais resiliente por não se basear em apenas um canal para a associação e reassociação de nós, ao custo de um maior atraso tanto para a entrada de novos nós, quanto para a reassociação dos nós caso se desconectem de um ponto da rede, dada a necessidade de varredura dos canais utilizados.

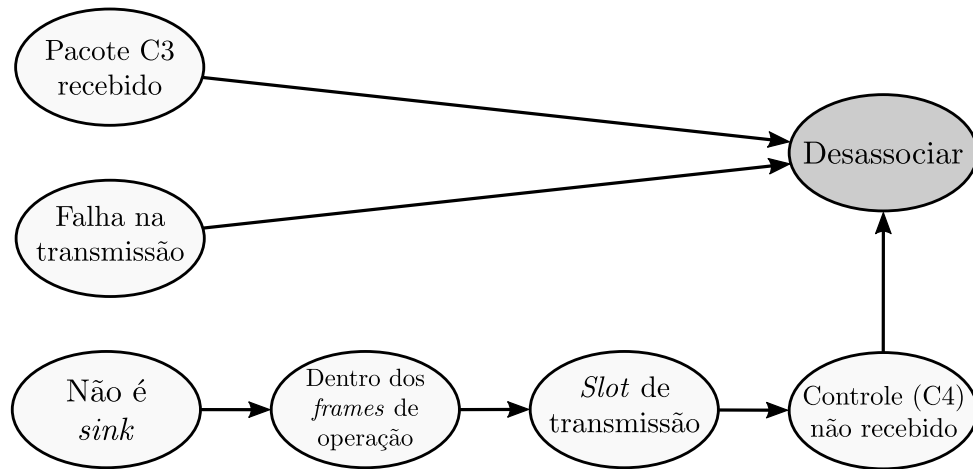
### 3.7 Desassociação

Além da associação, o protocolo também deve lidar com a desassociação dos nós. Para isso, uma série de condições determinam a desassociação de um nó. Quando isso ocorre, o nó retorna às suas configurações iniciais, descarta toda a informação que possui da rede e passa a tentar se associar novamente.

A Figura 3.8 mostra as condições para a desassociação de um nó. O primeiro caso ocorre quando o nó recebe um pacote de controle com o comando C3 (*C\_COMM\_DISCONNECT*) de seu nó pai. Para que um nó pai envie esse tipo de pacote para um nó filho são verificadas diversas condições, que serão analisadas posteriormente nesta seção.

A segunda condição para um nó filho se desassociar do nó pai é a falha na transmissão de algum pacote. Toda troca direta de mensagens entre nós pais e filhos consiste na transmissão de um pacote (de controle ou dados) e seu respectivo pacote de confirmação, como recomendado por Reiner, Fairhurst e Wood (2004). Caso o nó filho envie pacotes para o nó pai e não

Figura 3.8 – Condições testadas para determinar a desassociação.



Fonte: do autor (2017).

receba confirmação, a conexão é considerada instável e o filho se desassocia. Deve-se ressaltar que o Gen3 utiliza retransmissões, e que para uma transmissão ser considerada falha todas as tentativas foram esgotadas sem que se obtivesse confirmação de entrega do pacote.

Por fim, o último caso de desassociação de um nó envolve o teste de quatro condições ao final do *slot*. O primeiro teste verifica se o nó não é o *sink* ( $\{n_x \neq n_{sink} | n_x \in N\}$ ), pois o *sink* nunca se desconecta da rede. Em seguida, é verificado se o nó está dentro de seus *frames* de operação, determinado pela condição  $\{f \in F_x | x \in N\}$ . A próxima condição testada é se o *slot* atual é o *slot* de transmissão do nó, aquele em que ele se comunicaria com seu nó pai, dado por  $\{S = S_x^{tx}\}$ .

Se todas as condições anteriores forem verdadeiras, é verificado se o nó recebeu um pacote de controle com o comando C4 de seu nó pai. Em todo *frame* pertencente a um nó ocorrem pelo menos duas transmissões de mensagens de controle (com seus respectivos pacotes de confirmação). A primeira transmissão parte do filho para o pai sendo encerrada com um pacote de controle com o comando C4. Depois o pai transmite os pacotes que tenha em *buffer* para o filho e finaliza com um pacote de controle com o comando C4. Se por algum motivo este pacote de controle do pai não chegar ao filho as informações de sincronização são perdidas, além de não poder se garantir que o filho possua a configuração mais recente da rede.

Satisfeitas essas quatro condições, o nó decide pela sua desassociação de seu nó pai e volta a tentar entrar na rede novamente.

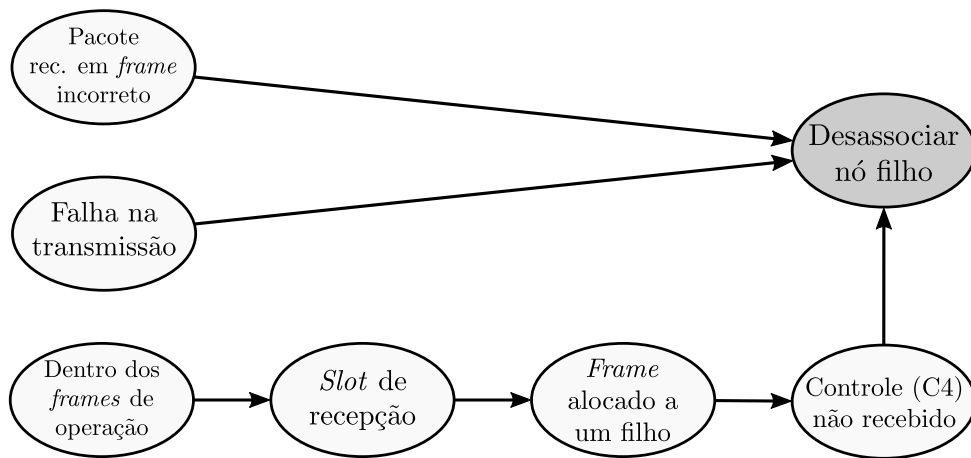
A Figura 3.9 apresenta as condições para que um nó desassocie um nó filho. Essa desassociação envolve tanto a transmissão de um pacote de controle com comando C3, quanto a

marcação dos *frames* ocupados por esse filho como livres, permitindo a realocação ou devolução desses *frames*. O pacote C3 não requer confirmação, pois pode ser transmitido para um nó filho que já tenha decidido pela sua desassociação.

O primeiro caso de desassociação de um nó filho é quando este envia um pacote fora de seus *frames* de operação (exceto quando solicitado via multiplexação estatística). Esse caso pode ocorrer se o filho estiver usando informações desatualizadas da rede, ou tenha perdido a sincronização. Nesse caso, após a primeira mensagem recebida no *frame* incorreto, o nó pai transmite um pacote C3 para o filho e executa as operações de remoção de *frames*.

O segundo caso é semelhante ao anterior quando há falha na transmissão de pacotes. Se durante a comunicação com um nó filho ocorrer a falha nas transmissões, mesmo utilizando os mecanismos de retransmissão, o nó considera a conexão instável e desassocia o filho. Ainda assim, o pacote C3 é transmitido para tentar garantir que o nó filho seja notificado da decisão de desassociação tomada por seu nó pai.

Figura 3.9 – Condições para envio de pacote de desconexão.



Fonte: do autor (2017).

Por fim, ao finalizar o *slot* corrente o nó pai analisa quatro condições. Na primeira, o pai verifica se está dentro de seus *frames* de operação ( $\{f \in Fx | x \in N\}$ ), em seguida checa se o *slot* atual era para comunicação com seus nós filhos, dado por  $\{S = S_x^{rx}\}$ . Depois é verificado se o *frame* atual estava alocado a algum nó filho, e finalmente verifica se não recebeu um pacote de controle com comando C4 durante esse *slot*. Caso todas as condições sejam verdadeiras, o nó filho alocado ao *frame* atual é desassociado e o respectivo pacote C3 é transmitido.

### 3.8 *Handoff e buffer*

Quando um nó se desconecta de um ponto da rede, é esperado que ele volte a se reconectar em outro ponto e continue sua operação. Para uma área totalmente coberta por uma infraestrutura de nós fixos, esse tempo de desassociação e reassociação (*Handoff*) depende apenas da capacidade do protocolo MAC. Quanto menor o tempo de *handoff*, menor o atraso na entrega dos pacotes no *buffer* dos nós fazendo *handoff* e, portanto, menor o impacto nas aplicações executadas pelo nó.

Uma das configurações que afetam diretamente o tempo de *handoff* é o uso ou não de CCC. Se configurado para usar CCC, o nó passa a ouvir no canal de controle imediatamente após a desassociação e caso contrário, é necessário varrer os canais em busca de pacotes de descoberta como descrito na Seção 3.6.

De acordo com Reiner, Fairhurst e Wood (2004), é desejável que o protocolo MAC não descarte pacotes silenciosamente durante um período de desconexão, mas sim que possua uma interface de comunicação com a camada superior que lhe permita recusar novos pacotes durante esse período e voltar a requisitar pacotes quando restabelecer a conexão. Se isso não for possível, a camada deve ser capaz de armazenar ao menos alguns pacotes até que a conexão seja restabelecida.

Por ser um protocolo escalonado, é imprescindível o uso de um *buffer* para reter pacotes até que o nó volte a ter oportunidade de transmissão. Caso contrário, a transmissão de dados dependeria da coincidência entre o momento de envio dos pacotes da camada superior e o momento em que o nó é escalonado para transmitir, o que tornaria inviável a operação do protocolo.

Quando o nó se desassocia mas ainda possui pacotes em *buffer*, estes são mantidos e transmitidos quando o nó se reconecta. Se mais dados forem enviados pela camada superior, eles são armazenados até que o espaço em *buffer* se esgote. Quando o nó finaliza o *handoff*, ao se reconectar à rede, os pacotes em *buffer* são transmitidos.

### 3.9 Transmissão de dados

Imediatamente após sua associação o nó já pode utilizar o restante do *slot* em que se associou para a transmissão de dados. Após finalizar a transmissão em um *slot*, o nó só terá outra chance quando estiver novamente em um de seus *frames*. Nós que possuem mais filhos

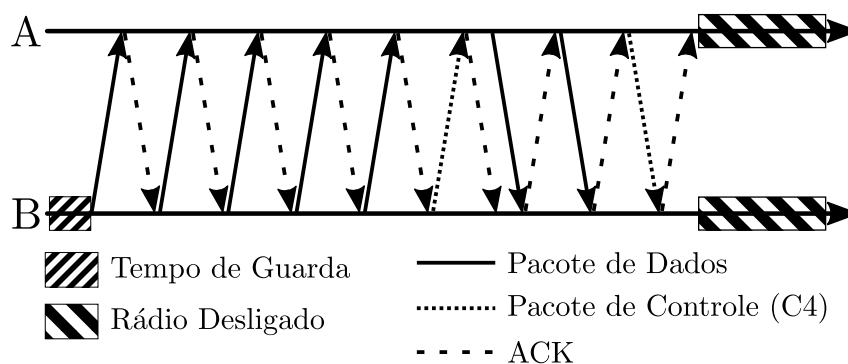
em sua sub-rede possuem mais *frames* e, logo, mais oportunidades de transmissão para o nó pai.

A comunicação é iniciada com a transmissão dos pacotes do nó filho para o nó pai e dois fatores são considerados para esse comportamento. O primeiro fator é que um nó pai possui mais *frames* que seus nós filhos e, portanto, tem mais oportunidades de descarregar os pacotes de seu *buffer*. Dessa forma, o nó pai geralmente terá mais espaço livre para armazenar novos pacotes do que o nó filho. O segundo fator é que em redes nas quais a aplicação realiza primariamente a coleta de dados, o maior fluxo de informação ocorrerá no sentido dos nós filhos para os nós pais, também aumentando a ocupação de *buffer* dos nós filhos.

O filho aguarda um tempo de guarda no início do *slot* antes de iniciar as transmissões, para evitar erros de sincronização. Ao fim do tempo de guarda, são buscados no *buffer* os pacotes que devem ser encaminhados ao nó pai. O filho transmite os pacotes de dados e finaliza com um pacote de controle com comando C4 (definido na Tabela 3.2), que indica ao nó pai que pode transmitir seus dados. O pai busca em seu *buffer* os pacotes destinados ao filho ou à sua sub-rede, os transmite e finaliza também com um pacote de controle com comando C4.

Esse pacote de controle do nó pai é crucial para a sincronização do nó filho, pois carrega a estimativa de tempo global da rede, além de informar modificações na configuração da rede. Esse pacote de controle do nó pai determina o fim das transmissões durante o *slot*, permitindo que ambos desliguem seus rádios para economia de energia. A comunicação entre um nó filho (B) e um nó pai (A) durante um *slot* é ilustrada na Figura 3.10.

Figura 3.10 – Comunicação entre nó filho (B) e nó pai (A) durante um *slot*.



Fonte: do autor (2017).

Um tempo de guarda no final dos *slots* também pode determinar o encerramento das transmissões caso estas ocupem todo o *slot*. Esse tempo de guarda é suficiente apenas para



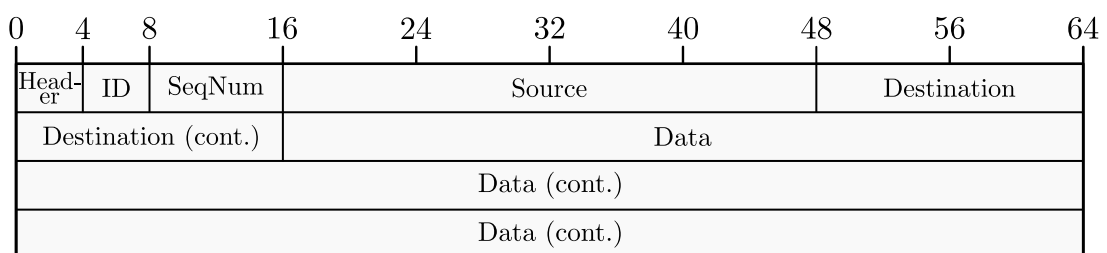
a transmissão dos pacotes de controle, permitindo a sincronização e a troca de configurações dentro do *slot*.

Os pacotes são transmitidos um por vez com confirmação (ACK) a cada pacote, seguindo o comportamento do transceptor nRF24l01+ (NORDIC\_SEMICONDUCTOR, 2013). O modo de operação desse transceptor foi escolhido por ele ter sido utilizado anteriormente em trabalhos que serviram de base para o desenvolvimento deste protocolo (JÚNIOR; MARQUES; CORREIA, 2016).

O transmissor envia um pacote e aguarda um período predeterminado pelo ACK, e se durante este período não recebê-lo, é realizada uma nova tentativa. No nRF24l01+ o período de espera por ACK pode ser configurado entre 250  $\mu$ s e 4.000  $\mu$ s e a quantidade máxima de tentativas de retransmissão entre 0 e 15. Neste trabalho foi usado um intervalo de 750  $\mu$ s e um máximo de 5 tentativas de retransmissão. Essa configuração vai de encontro às recomendações em RFC3819 (REINER; FAIRHURST; WOOD, 2004) e RFC3366 (FAIRHURST; WOOD, 2002) de que: (i) em redes sem fio com pacotes pequenos (< 100 bytes) e relativamente alta taxa de erros, mecanismos de controle de erros podem ajudar a melhorar o desempenho; (ii) a persistência do mecanismo de recuperação usado na camada de enlace deve ser reduzida, de modo que o protocolo não fique estagnado para tentar garantir a entrega de um único pacote, i.e. a quantidade de tentativas de retransmissão deve ser limitada e alguma ação deve ser tomada quando esta se esgota, como a notificação da camada superior.

Além das características de transmissão baseadas no nRF24l01+, os pacotes de controle e de dados são limitados a 32 bytes por ser o tamanho máximo de carga que o transceptor suporta. O formato do pacote de controle foi descrito na Seção 3.4 e o de dados tem o formato descrito na Figura 3.11. O pacote tem os campos de cabeçalho (*Header*) e *ID* de 4 bits, um campo de número de sequência (*SeqNum*) de 8 bits, campos de endereço de origem e destino (*Source* e *Destination*) de 32 bits e os demais 176 bits para dados vindos da camada superior.

Figura 3.11 – Formato do pacote de dados.



Fonte: do autor (2017).

### 3.10 Multiplexação estatística coordenada - MXC

Em protocolos TDMA os nós operam em tempos bem definidos e muitas vezes precisam aguardar por longos períodos até terem outra oportunidade de transmissão. Durante esse período de espera a aplicação pode gerar dados e esgotar a capacidade de *buffer* do nó, implicando em descarte de pacotes. Além disso, o tempo útil para transmissão de dados dentro de um *slot* pode não ser suficiente para descarregar todos os pacotes em *buffer*, contribuindo para esgotar sua capacidade.

O mecanismo de Multiplexação Estatística Coordenada (MXC) visa dinamizar essa operação permitindo a utilização oportunística de *slots*. Ao iniciar o *slot*, o nó filho transmite pacotes de dados para o nó pai e encerra com um pacote de controle do tipo C4. O nó pai, ao receber esse pacote de controle, transmite os dados em *buffer* destinados ao nó filho e também encerra com um pacote de controle C4. No campo *Child MX Request*, o nó pai especifica o endereço de algum outro filho para que ele utilize o restante desse *slot* para transmitir dados.

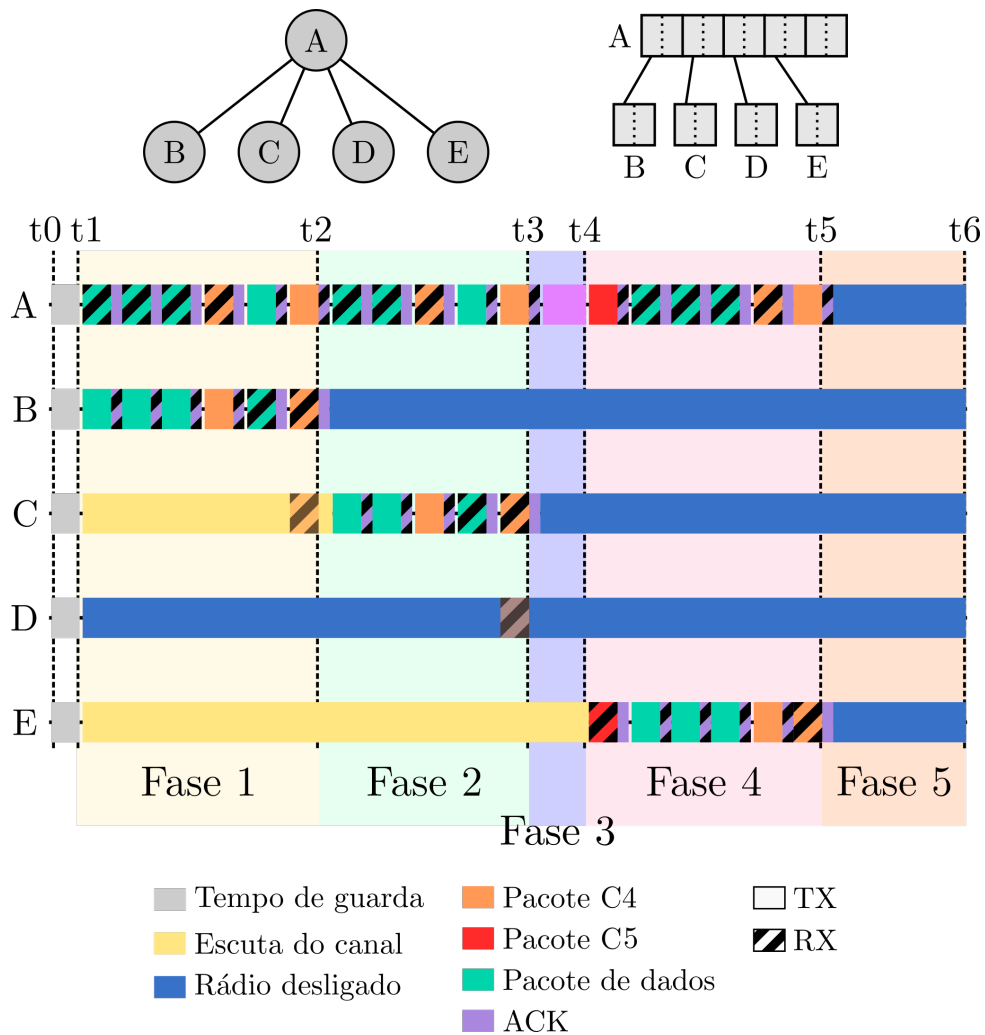
Para que o nó filho seja capaz de aproveitar o restante dos *slots* de outro nó, ele deve se manter com o rádio ativado, configurado para o canal de comunicação com o nó pai e precisa capturar pacotes destinados ao outro nó. Isso pode aumentar o consumo energético do nó por precisar ouvir fora de seus períodos predefinidos, e ainda assim não ter oportunidade de transmissão, caso outros nós ocupem todo o *slot*. Para atenuar esse problema o Gen3 permite a configuração de um limiar de ocupação de *buffer* para que o nó ative a MXC. Caso a ocupação de *buffer* esteja acima do valor definido, os nós ouvem fora de seus *slots* aguardando uma oportunidade de transmissão, caso esteja abaixo do valor o nó desliga seu rádio para economizar energia.

Se o nó filho especificado pelo pai não estiver com a MXC ativa, devido à baixa ocupação de *buffer*, o nó pai poderia ficar o restante do *slot* aguardando pelas transmissões e o restante *slot* seria desperdiçado. Para evitar esse problema o nó pai define um limite de tempo para que esse filho responda. Se não receber resposta dentro desse tempo, o nó pai seleciona o próximo filho e faz uma requisição explícita de MXC, com o pacote de controle de comando C5. Se o próximo filho selecionado for novamente o dono do *slot*, nenhuma mensagem é enviada e o nó pai desativa seu rádio.

O tempo que o nó pai aguarda para fazer a requisição explícita de MXC depende das configurações de retransmissão. Seja  $R$  a quantidade máxima de retransmissões que um nó faz e  $I$  o intervalo entre as retransmissões, o tempo de espera pela resposta do nó filho será de  $R \times I$ .

A Figura 3.12 ilustra o funcionamento da MXC em uma topologia contendo 5 nós, sendo *A* o nó *sink* com os 4 nós filhos *B*, *C*, *D* e *E*. No topo da figura é mostrada a topologia formada e como ficaria a alocação de *frames* nessa rede. A figura ilustra o tráfego em um *slot* quando a MXC está sendo utilizada. O *slot* em questão é o primeiro *slot* do primeiro *frame*, no qual ocorre a comunicação entre o nó *B* e o nó *A*.

Figura 3.12 – Transmissões utilizando MXC.



Fonte: do autor (2017).

O tempo compreendido entre  $t_0$  e  $t_1$  é referente ao tempo de guarda, em que os nós não executam nenhuma atividade. Após  $t_1$ , destacado como Fase 1, ocorre a comunicação entre *B* e *A*. O nó *B* transmite 3 pacotes de dados para *A*, que responde com pacotes ACK para cada um. *B* finaliza suas transmissões para *A* com um pacote de controle C4, que o nó *A* responde com um ACK e então transmite um pacote de dados para *B*. Ao receber este pacote, o nó *B* desliga seu rádio para economizar energia.

No pacote C4 enviado para *B*, o nó *A* especifica o endereço do nó *C* como próximo filho a transmitir por MXC e informa quanto de espaço ainda tem em *buffer*. O nó *C*, que até então permanecia escutando as transmissões, detecta este pacote e aguarda um período igual ao intervalo de retransmissão para iniciar o envio de pacotes para *A*. De  $t_2$  a  $t_3$ , destacado como Fase 2, *C* envia 2 pacotes para *A* e finaliza com um pacote C4. *A* transmite um pacote de dados para *C* e finaliza com um pacote C4, agora informando que o próximo filho selecionado para MXC é o nó *D*.

Por estar com baixa ocupação de *buffer*, *D* está desde o início do *slot* com seu rádio desligado e não detecta o pacote C4 transmitido por *A*. No período de  $t_3$  a  $t_4$ , destacado como Fase 3, o nó *A* aguarda pela resposta do nó *D* pelo tempo predefinido de  $R \times I$ , descrito anteriormente.

Encerrado o período de espera, no intervalo de  $t_4$  a  $t_5$  destacado como Fase 4, *A* transmite um pacote C5 com uma requisição explícita de MXC para o nó *E*, que permanecia ouvindo transmissões desde o início do *slot*. Ao receber essa requisição, *E* transmite 3 pacotes para *A*, encerrando com o pacote C4. *A* por sua vez não tem pacotes de dados endereçados a *E* e transmite apenas um pacote C4 indicando o fim de transmissões. Com isso, *E* pode desligar seu rádio pelo restante do *slot*.

Na Fase 5, de  $t_5$  a  $t_6$ , o nó *A* verifica que todos os seus filhos já transmitiram nesse *slot*. O próximo filho selecionado seria novamente o nó *B*, mas como ele é o dono do *frame* atual e já transmitiu nesse *slot*, *A* simplesmente desliga seu rádio para economizar energia. É assumido que caso *B*, dono do *frame* e do *slot* atuais, tivesse mais dados para transmitir, essas transmissões teriam ocorrido durante a Fase 1. Além disso, após receber um pacote C4 do nó pai, um nó filho deve obrigatoriamente encerrar suas transmissões durante o *slot*.

A seleção de próximo filho para MXC segue um padrão circular com duas restrições. A primeira define que o nó selecionado para MXC não pode ser o dono do *frame*, pois ele já teve oportunidade de transmitir no *slot* atual. A segunda determina que um nó é selecionado para MXC apenas uma vez por *slot*. Por seguir um padrão circular, o último filho selecionado no *frame* passado também será o último selecionado no *frame* atual.

### 3.11 Premissas e limitações

Para que o Gen3 opere de forma correta algumas premissas são consideradas:

1. O rádio utilizado nos nós não opera como transmissor e receptor ao mesmo tempo. Se configurado como receptor, o rádio pode apenas receber pacotes, e para que possa

transmitir dados é necessário que saia do modo receptor e se configure como transmissor. Além disso, o rádio opera em apenas um canal por vez, no entanto, canais adjacentes podem causar interferências entre si, como ocorre em ambientes reais.

2. A rede possui apenas um *sink*. O Gen3 assume que existe apenas um nó *sink* operando em uma dada região e utilizando o mesmo tipo de rádio. Caso existam dois *sinks* em distância de interferência e usando rádios que operam na mesma frequência, uma coordenação adicional dos canais será necessária para que não ocorra interferência entre as diferentes redes.
3. O tempo de duração dos *slots* é suficiente para abranger: (i) os tempos de guarda; (ii) o tempo de CCA utilizado na associação; (iii) a transmissão de 2 pacotes de controle e seus respectivos pacotes de confirmação; (iv) a transmissão de pelo menos um pacote de dados e seu pacote de confirmação.
4. O tempo de guarda no início dos *slots* é maior que o erro de sincronização. Nas simulações os nós têm seu tempo sincronizado desde o início das execuções, porém, neste trabalho os tempos de guarda para superar erros de sincronização são embasados em resultados reais obtidos por outros autores, discutidos no Capítulo 2.

Além dessas premissas, existem também algumas limitações do Gen3. Da forma como o protocolo foi implementado, tomando como referência o rádio nRF24101+, o tamanho dos pacotes é de no máximo 32 Bytes. Com isso, alguns campos tiveram tamanhos restritos para que todas as informações necessárias fossem transmitidas em apenas um pacote de controle.

A quantidade de *frames* é limitada a 255, pois o campo *Network Frame Count* do pacote de controle tem apenas 8 bits, permitindo uma representação de valores de 0 a 254. Isso implica na limitação da quantidade de nós dessa rede em no máximo 255 nós, dado que deve-se ter um *frame* para cada nó. Deve-se ressaltar que com o uso de rádios que suportem pacotes maiores esse valor pode ser aumentado, permitindo também uma maior escalabilidade da rede.

Outra limitação do Gen3 ocorre devido ao endereçamento. Utilizando endereços de 4 Bytes e a máscara descrita na seção 3.5, é possível que a rede tenha no máximo 7 saltos. Como o simulador permite maior manipulação dos pacotes transmitidos, foi possível utilizar endereços de 6 Bytes, suportando até 11 saltos, permitindo uma melhor análise da escalabilidade do protocolo. Com a modificação do formato de máscaras é possível aumentar a quantidade de

saltos suportada pelo Gen3, ao custo da redução da quantidade máxima de filhos que um nó pode ter.

### 3.12 Considerações finais

Para lidar com variados tipos de tráfego com comportamento consistente, o Gen3 se baseia em TDMA. A divisão de tempo é feita em ciclos, que são divididos em vários *frames*. A quantidade de *frames* é igual à quantidade de nós, e cada *frame* divide-se em dois *slots*. Para lidar com erros de sincronização e garantir a correta configuração dos nós para a comunicação, são usados tempos de guarda no início e no fim dos *slots*.

Pela forma de alocação de *frames* é garantido que nós filhos de um mesmo pai trabalhem em períodos diferentes, bem como toda a sub-rede de cada um, sem causar colisões. A configuração de *slots* de transmissão e canais evita colisões entre nós que compartilham *frames* em uma dada sub-rede. A quantidade de *frames* é dinâmica, determinada pela quantidade de nós associados, e é corrigida no caso de saída de nós. Com isso, é sempre mantida a quantidade correta de *frames*.

Um nó recém associado à rede já pode iniciar a transmissão de dados sem que seja necessário esperar pela reorganização da rede devido à sua entrada. Para acelerar o processo de associação pode-se configurar o protocolo para usar CCC, de modo que os nós troquem pacotes de associação em um canal comum. Outra forma possível é sem o uso de CCC, de modo que os nós tentando se associar tenham que escutar em todos os canais da rede buscando os pacotes de associação. Apesar de dificultar a associação de novos nós, na configuração sem CCC o protocolo não fica dependente de apenas um canal.

Para melhor explorar o espectro, um mecanismo de multiplexação estatística coordenada (MXC) é utilizado, permitindo que os nós utilizem o restante dos *slots* de outros nós associados ao mesmo pai. A MXC é ativada em um nó com base na taxa de ocupação de seu *buffer*, permitindo que nós gerando pouco tráfego economizem energia enquanto nós com alta geração de dados priorizem a transmissão dos dados ao custo de um maior consumo energético.

No próximo capítulo será descrita a metodologia utilizada para avaliação do protocolo e sua comparação com outras propostas.

## 4 METODOLOGIA DE AVALIAÇÃO

Esta Seção apresenta a metodologia utilizada para avaliação do trabalho, como foram realizados os testes, a coleta dos dados, as topologias e modelos de movimento utilizados, os parâmetros e as configurações das simulações. Para a realização dos testes foi utilizado o simulador OMNeT++ (VARGA et al., 2001) com o *framework* INET, que possuem modelos realistas de interferência, ruído e propagação de sinal para a avaliação de redes sem fio.

### 4.1 Parâmetros de transmissão do protocolo

Os parâmetros relacionados ao rádio foram definidos de acordo com as características do transceptor nRF24l01+. A potência de transmissão foi definida em 0 dBm (1mW), com taxa de transmissão de 2 Mbps, largura de banda de 2 MHz e sensibilidade de -82 dBm. Para o meio de transmissão foi utilizado ruído base de -90 dBm e relação sinal-ruído e interferência (SNIR) de 4 dB. Com esses parâmetros, obteve-se um raio de alcance de 150 metros para cada nó.

Alguns parâmetros do Gen3 foram mantidos fixos em todos os testes, sendo eles: duração de *slot* de 20 ms e tempo de guarda de 1 ms, *buffer* com capacidade para 100 pacotes, intervalo entre retransmissões de 750  $\mu$ s e um máximo de 5 tentativas de retransmissão por pacote. Outros parâmetros do protocolo, como uso de CCC ou não, uso de MXC e taxa de ocupação de *buffer* para ativação de MXC foram definidos separadamente para cada teste.

O tempo de guarda de 1 ms no início dos *slots* foi definido com base nas informações obtidas sobre protocolos de sincronização, discutidos na Seção 2.2.2. No final dos *slots* foi utilizado 2 ms de tempo de guarda, resultando em uma janela de transmissão de 17 ms dentro de cada *slot*.

A Tabela 4.1 sumariza os parâmetros do protocolo e do rádio.

### 4.2 Topologias

Para a avaliação e comparação do Gen3 foram utilizadas diferentes topologias físicas, que determinam como é a disposição dos nós pelo espaço. As topologias são divididas em duas categorias: estáticas e mistas. Não foram utilizadas topologias em que todos os nós possuem mobilidade devido à possibilidade de particionamento da rede. O Gen3 não lida com o particionamento da rede e os nós dependem da conectividade com o *sink* (direta ou *multihop*) para operarem.

Tabela 4.1 – Parâmetros de transmissão e do protocolo Gen3.

Parâmetro	Valor
Duração do <i>slot</i>	20 ms
Tempo de guarda A	1 ms
Tempo de guarda B	2 ms
Capacidade de <i>buffer</i>	100 pacotes
Tentativas de retransmissão	5
Intervalo de retransmissão	750 $\mu$ s
Potência de transmissão	0 dBm (1 mW)
Taxa de transmissão	2 Mbps
Largura de banda	2 MHz
Frequências do rádio	2.400 MHz a 2.525 MHz
Sensibilidade do rádio	-82 dBm
Ruído base	-90 dBm
SNIR	4 dB
Distância de alcance	150 m

Fonte: do autor (2017).

#### 4.2.1 Topologias estáticas

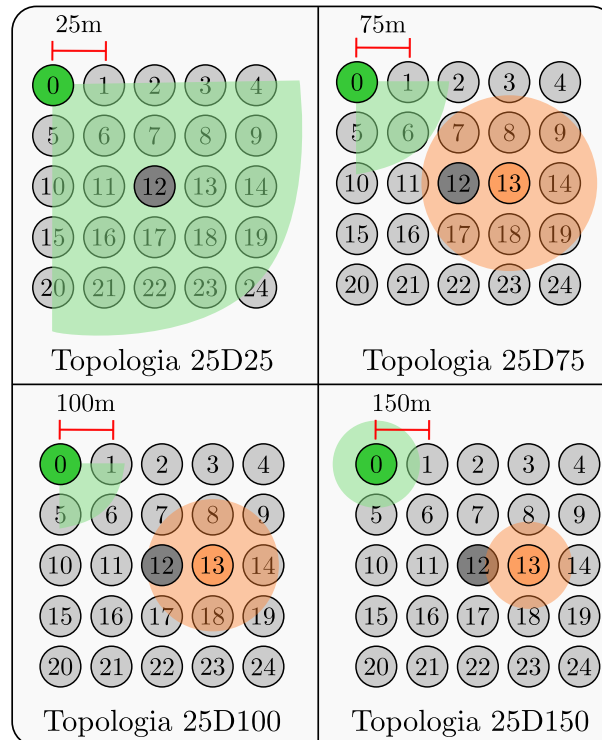
Nas topologias estáticas nenhum nó possui mobilidade e todos continuam na mesma posição durante toda a simulação. Foram utilizadas topologias com 25 nós dispostos de forma regular pelo espaço e com o nó central como *sink*, como seria disposta uma RSSF monitorando caixas ou galões armazenados em uma planta industrial. Além disso, foram avaliadas topologias lineares, uma com 12 nós com o *sink* em uma extremidade, e outra com 23 nós e o *sink* no ponto central, como exemplo de uma RSSF monitorando torres em uma linha de transmissão de energia.

A Figura 4.1 mostra as quatro topologias de 25 nós com disposição regular, todas tendo o nó central (nó 12) como *sink*. Na topologia 25D25, a distância vertical e horizontal dos nós é de 25 metros, permitindo o alcance de todos para todos os nós. Na figura, a parte sombreada em verde indica o raio de alcance do nó 0, ilustrando que ele é capaz de alcançar todos os nós. Isso é válido para todos os demais. Essa topologia permite avaliar o comportamento do protocolo em uma rede densa, na qual a comunicação de cada nó pode interferir na comunicação dos demais.

Na topologia 25D75 a distância entre os nós é de 75 metros, fazendo com que um nó alcance até quatro vizinhos na horizontal e na vertical, e um vizinho na diagonal. Isso permite avaliar o protocolo em uma rede menos densa que a da topologia 25D25. Na 25D100, com espaçamento de 100 metros entre cada nó, o alcance dos nós é de um vizinho em cada direção e um na diagonal. Por fim, na topologia 25D150, com a distância entre os nós igual ao raio



Figura 4.1 – Topologias regulares com 25 nós e nó 12 como *sink*.



Fonte: do autor (2017).

de alcance do rádio, cada nó consegue se comunicar apenas com os vizinhos mais próximos na horizontal e na vertical, sendo a topologia mais esparsa.

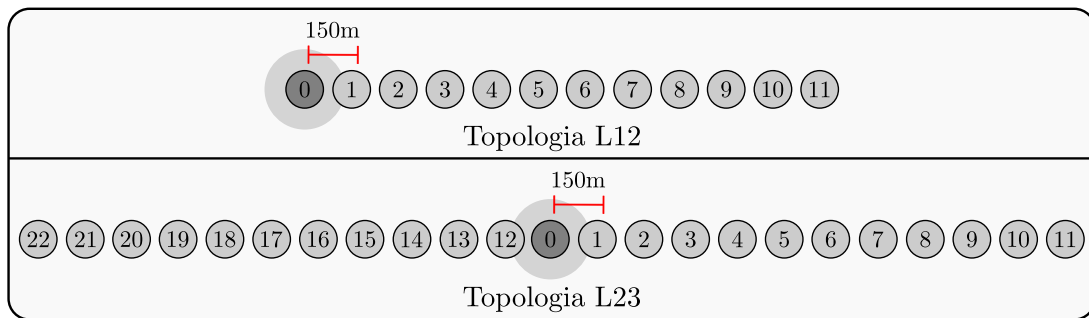
Essas topologias permitem avaliar a operação do protocolo de acordo com a densidade de nós. A escolha de uma disposição regular visa evitar variações nos resultados devido à disposição aleatória de nós, permitindo a comparação dos resultados tendo como variáveis apenas os parâmetros do protocolo.

A Figura 4.2 apresenta as duas topologias lineares. A topologia L12 tem 12 nós e o *sink* (nó 0) situado na extremidade. Já a topologia L23 possui 23 nós com o *sink* central. Em ambas a distância entre os nós é de 150 metros, tal que cada um só se comunique com aqueles imediatamente adjacentes.

#### 4.2.2 Topologias mistas

Nas topologias mistas alguns nós são estáticos, fornecendo conectividade sobre uma área por onde os demais nós se movem. Os nós estáticos iniciam sua operação antes dos nós móveis, para que a rede esteja totalmente configurada para a entrada dos nós móveis.

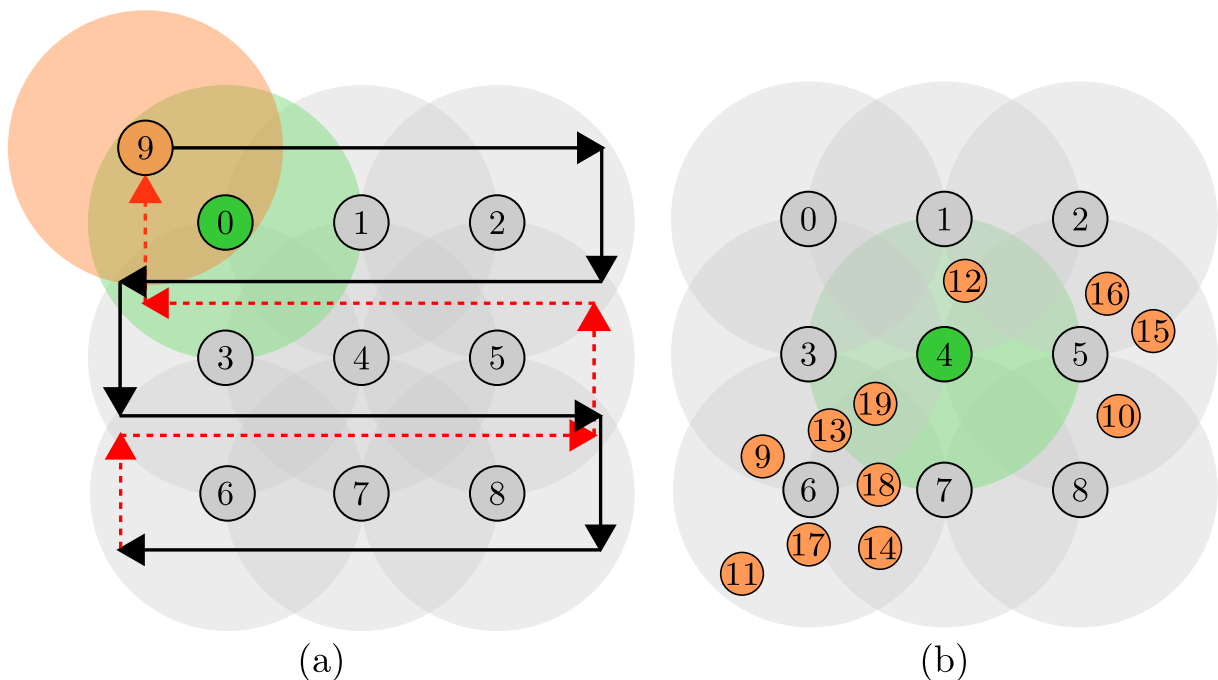
Figura 4.2 – Topologias lineares com 12 e 23 nós.



Fonte: do autor (2017).

A primeira topologia mista, denominada 9F1, é apresentada na Figura 4.3a, tendo 9 nós fixos e 1 móvel. Os nós de 0 a 8 ficam fixos cobrindo uma área, enquanto o nó 9 possui mobilidade controlada, cujo caminho é definido pelas setas. O objetivo dessa topologia é avaliar o comportamento do protocolo no nó móvel, tendo em vista que sempre estará na área de cobertura de algum dos nós fixos. Com isso, a capacidade do nó de se desconectar de um ponto e reconectar em outro ponto depende diretamente da capacidade de reorganização e reassociação do protocolo, eliminando casos em que o nó estivesse fora da área de cobertura.

Figura 4.3 – Topologia 9F1 - 9 nós fixos e 1 nó com mobilidade controlada.



Fonte: do autor (2017).

A segunda topologia mista, denominada 9F11, é ilustrada na Figura 4.3b e consiste em 9 nós fixos (0 a 8), tendo o nó 4 como *sink* e 11 nós (9 a 19) com mobilidade aleatória. Os nós

fixos cobrem uma área, por onde os nós móveis se movimentam aleatoriamente. Parâmetros adicionais como a velocidade e a direção dos nós são definidos de forma independente em cada cenário de teste, e serão apresentados posteriormente.

### 4.3 Sumário das topologias de teste

A Tabela 4.2 sumariza as topologias utilizadas nos testes. A primeira coluna traz os nomes dados às topologias, que serão utilizados para referenciá-las no restante do trabalho e a segunda coluna traz uma breve descrição de cada uma.

Tabela 4.2 – Sumário das topologias utilizadas nos testes.

Nome	Descrição
25D25	Regular, 25 nós com todos no alcance de todos.
25D75	Regular, 25 nós, alcance de 2 vizinhos na horizontal, 2 na vertical e 1 na diagonal.
25D100	Regular, 25 nós, alcance de 1 vizinho em cada direção e 1 na diagonal.
25D150	Regular, 25 nós, alcance apenas dos vizinhos mais próximos em cada direção e nenhum na diagonal.
L12	Linear, 12 nós com <i>sink</i> na extremidade.
L23	Linear, 23 nós com <i>sink</i> na posição central.
9F1	9 nós fixos cobrindo uma área e 1 nó com mobilidade controlada pela área de cobertura.
9F11	9 nós fixos cobrindo uma área e 11 nós com mobilidade aleatória pela área de cobertura.

Fonte: do autor (2017).

### 4.4 Padrões de tráfego

Diferentes padrões de tráfego foram adotados para os testes a fim de caracterizar o comportamento do Gen3. O primeiro padrão de tráfego é baseado na geração constante de dados (CBR - *Constant Bit Rate*), no qual uma aplicação gera pacotes em intervalos fixos. Foram utilizados intervalos de 30, 25, 20, 15, 10 e 5 milissegundos de geração de pacotes, resultando em taxas de respectivamente 33, 40, 50, 66, 100 e 200 pacotes por segundo por cada nó.

O segundo padrão de tráfego é baseado em rajadas no qual a aplicação gera uma quantidade de pacotes em intervalos aleatórios. Neste trabalho, o padrão de rajadas é denominado VBR (*Variable Bit Rate*) e é controlado por três parâmetros:

1. **Intervalo entre pacotes:** esse parâmetro determina o intervalo de transmissão entre cada pacote durante uma rajada, sendo utilizado um valor aleatório entre 1 ms e 5 ms para esse intervalo a cada pacote transmitido.
2. **Duração da rajada:** determina durante quanto tempo cada nó ficará transmitindo pacotes. Para esse parâmetro é utilizado um valor aleatório entre 5 ms e 100 ms, determinado a cada nova rajada.
3. **Intervalo entre rajadas:** o intervalo entre as rajadas é um valor aleatório entre 500 ms e 5 s, determinado a cada intervalo.

Aproveitando a capacidade do Gen3 de encaminhar pacotes para qualquer nó da rede, a definição dos endereços de destino dos pacotes em uma rajada também é aleatória. A cada nova rajada os nós determinam um novo endereço de destino para os pacotes da rajada em execução, permitindo avaliar não só a coleta de dados pelo *sink*, mas também a comunicação independente entre todos os nós da rede.

A aplicação gera tráfego transmitido via UDP (*User Datagram Protocol*), para o qual não existe confirmação fim-a-fim. Um protocolo semelhante ao protocolo ARP (*Address Resolution Protocol*) determina o endereço MAC para encaminhamento do pacote a partir do endereço IP de destino. Pela alocação dinâmica dos endereços durante a associação, e pela sua organização em árvore, o Gen3 é capaz de encaminhar pacotes através da rede sem a necessidade de um protocolo de roteamento, apenas conhecendo o endereço de destino final de cada pacote.

Todo nó associado é capaz de transmitir dados pela rede. Enquanto desconectados, o nós armazenam os pacotes recebidos da camada superior até esgotar o *buffer*. O início das transmissões varia de acordo com a topologia e seu tempo de estabilização médio, sendo definido de forma independente em cada cenário de teste. Os pacotes transmitidos têm tamanho fixo de 32 Bytes, que é o tamanho máximo suportado pelo rádio nRF24l01+.

#### 4.5 Cenários e métricas

Os cenários de testes combinam configurações do protocolo MAC, topologias físicas e padrões de tráfego. Em cada cenário são avaliados conjuntos diferentes de métricas. Para a descrição dos cenários de testes será utilizado o modelo apresentado na Figura 4.4. O elemento

mais à esquerda dá o nome do teste. Em seguida são descritos os parâmetros que são mantidos fixos nos testes realizados.

Figura 4.4 – Formato de descrição de cenários.

Nome do teste - Parâmetro fixo		Parâmetro variável 1	Parâmetro variável 2	Topologias utilizadas	Métricas avaliadas
Nome do teste - Parâmetro fixo		Valor 1	[ Valor 1 Valor 2	[ Topologia 1 Topologia 2 ... Topologia N	[ Métrica 1 ... Métrica 2
		Valor 2	[ Valor 1 Valor 2		

Fonte: do autor (2017).

Cada parâmetro variável gera uma bifurcação e os valores são apresentados em seguida. Após a descrição dos parâmetros são listadas as topologias que foram avaliadas nos testes, seguindo a nomenclatura apresentada anteriormente na Tabela 4.2. Por fim, o último elemento traz as métricas avaliadas, combinando os parâmetros descritos e as topologias utilizadas.

Esse esquema de descrição dos testes serve como guia para o acompanhamento dos resultados e os parâmetros apresentados não são exaustivos. As configurações adicionais de cada teste são detalhadas na descrição dos resultados.

O primeiro teste foi realizado para determinar o tempo médio de associação dos nós e de estabilização da rede, e é descrito na Figura 4.5. Para o teste de *Configuração* não foi utilizada MXC, mas o protocolo foi avaliado com e sem CCC. Foram testadas todas as topologias estáticas, de modo a eliminar a influência da movimentação dos nós nas métricas avaliadas.

Figura 4.5 – Cenários de avaliação de associação e estabilização.

Configuração - S/MXC	CCC	[ 25D25 25D75 25D100 25D150 L12 L23	[ Tempo de Associação Tempo de Estabilização
	S/CCC		

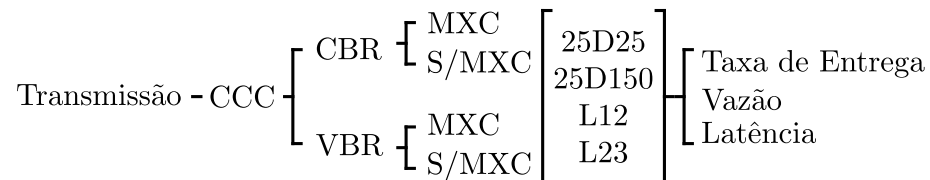
Fonte: do autor (2017).

Para a avaliação desse teste foram definidas duas métricas, *Tempo de Associação* e *Tempo de Estabilização*. A primeira indica o tempo médio necessário para que um nó se associe a uma rede recém inicializada. A segunda indica o tempo médio necessário para que essa rede finalize sua configuração, dado pelo momento em que a rede passa a utilizar a configuração com a quantidade de *frames* igual à quantidade de nós.

O segundo teste avalia a transmissão de dados e tem a configuração apresentada na Figura 4.6. Em todos os testes foi utilizado CCC, padrões de tráfego CBR e VBR, e para cada

padrão de tráfego o protocolo foi avaliado com e sem MXC. Foram utilizadas as topologias 25D25 e 25D150, permitindo avaliar o comportamento em redes densas e esparsas, e as topologias lineares, L12 e L23. Nos testes de transmissão são avaliadas as métricas de latência, vazão e taxa de entrega.

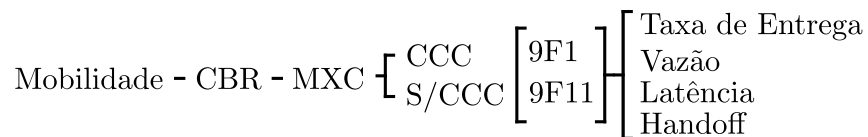
Figura 4.6 – Cenários de avaliação de transmissão de dados.



Fonte: do autor (2017).

O cenário descrito na Figura 4.7 foi utilizado para avaliar o protocolo com nós móveis. Para este cenário foi utilizado apenas o padrão de tráfego CBR e configuração com MXC. O foco desse teste é na avaliação do impacto do uso de CCC nas métricas de latência, vazão, taxa de entrega e tempo de *handoff*.

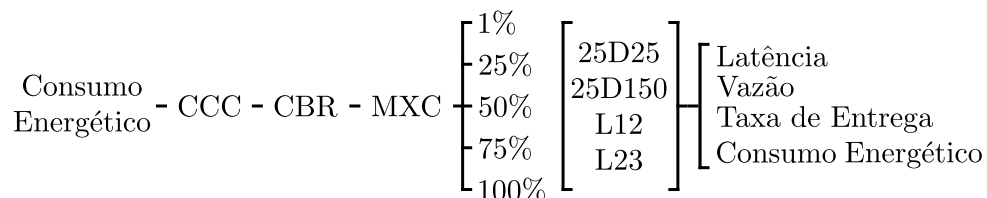
Figura 4.7 – Cenários de avaliação de mobilidade.



Fonte: do autor (2017).

Por fim, a Figura 4.8 apresenta a definição dos testes de consumo energético. Nesses testes foi utilizada a configuração com CCC e padrão de tráfego CBR em todos os casos. Por exigir que os nós monitorem o meio de transmissão por mais tempo, foram avaliados diferentes limiares para que os nós entrem em MXC. Apenas os nós com ocupação de *buffer* acima deste limiar continuarão ouvindo o meio de transmissão para tentar utilizar MXC.

Figura 4.8 – Cenários de avaliação de consumo energético.



Fonte: do autor (2017).

Os valores foram definidos em 1%, 25%, 50%, 75% e 100% de ocupação de *buffer* para que os nós façam MXC. Nesses testes são avaliadas as métricas de latência, vazão, taxa de entrega e consumo energético.

## 4.6 Protocolo para comparação

Para comparação com o Gen3 foi selecionado um protocolo dentre os analisados no Capítulo 2. A seleção se deu com base em três fatores:

- Semelhança do protocolo com o Gen3, focando em protocolos TDMA que suportassem alterações de topologia e que utilizassem múltiplos canais de comunicação simultaneamente para aumentar sua capacidade.
- Resultados obtidos pelos autores, com foco em trabalhos que apresentassem protocolos capazes de atingir altas taxas de entrega, alta vazão e baixa latência.
- Detalhamento da descrição do protocolo e disponibilidade de código fonte. Diversos trabalhos propõem protocolos mas com pouca descrição de seus mecanismos e implementação. Além disso, há uma grande dificuldade em encontrar o código-fonte desses protocolos, seja para simuladores ou para plataformas reais, dificultando a implementação dessas propostas para comparação.

Considerando os fatores avaliados, foi selecionado o protocolo MC-LMAC (INCEL et al., 2011). O MC-LMAC opera em TDMA e utiliza múltiplos canais de comunicação. Os autores compararam o MC-LMAC ao MMSN (ZHOU et al., 2006), com uma versão modificada do LMAC (HOESEL; HAVINGA, 2004) para operar em *clusters* em canais diferentes, e com o protocolo CSMA (LAM, 1980) sem mecanismos de RTS/CTS. O MC-LMAC superou os demais protocolos na maioria dos testes apresentados pelos autores.

### 4.6.1 Descrição do MC-LMAC e modificações realizadas

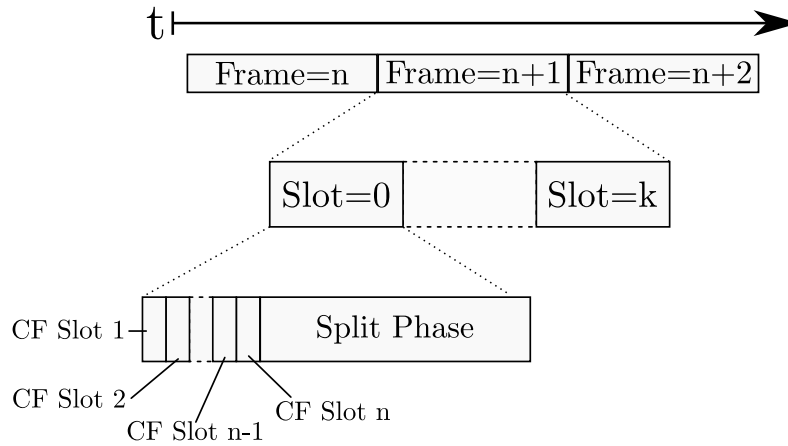
O *Multi-Channel Lightweight MAC* é uma extensão do protocolo LMAC para suportar sua operação em multicanal. Nesta seção serão apresentados os principais conceitos do MC-LMAC, sua operação e algumas modificações feitas na implementação utilizada neste trabalho.

#### Divisão de tempo

O formato de divisão de tempo do MC-LMAC é mostrado na Figura 4.9. O tempo é dividido em *frames* e a alocação para os nós é feita em *slots*, com cada nó tendo um *slot* para sua operação. Cada *slot* é dividido em duas fases, *CF Phase* e *Split Phase*. A *CF Phase* é composta por *CF Slots*, sendo a quantidade de *CF Slots* igual à quantidade de canais que o

protocolo irá utilizar. A *Split Phase* ocupa o restante do *slot*, onde podem ocorrer transmissões de dados em canais diferentes simultaneamente.

Figura 4.9 – Divisão de tempo adotada pelo MC-LMAC.



Fonte: adaptado de (INCEL et al., 2011).

### Seleção de *slot* e canal

A decisão do conjunto de operação (*slot*, canal) que o nó trabalha é crucial para que o nó possa transmitir dados. A quantidade de *slots* em um *frame* é fixa e previamente determinada com base na densidade esperada da rede. Se a quantidade de *slots* for inferior ao necessário, alguns nós podem nunca encontrar um conjunto de operação, sendo incapazes de se associar e transmitir dados. Já com uma quantidade de *slots* superior à necessária, o *frame* tem duração maior que o ideal, o que impacta na capacidade do protocolo em relação a latência, vazão e taxa de entrega.

Após coletar as informações dos vizinhos, o nó seleciona seu *slot* e seu canal de operação respeitando a duas condições: (i) o nó não pode selecionar nenhum *slot* ocupado por algum vizinho; (ii) *slots* ocupados por vizinhos a dois saltos podem ser selecionados, desde que em canal diferente. Dessa forma podem ocorrer transmissões simultâneas no mesmo *slot* em diferentes pontos da rede, mas em canais diferentes.

### Transmissão de dados

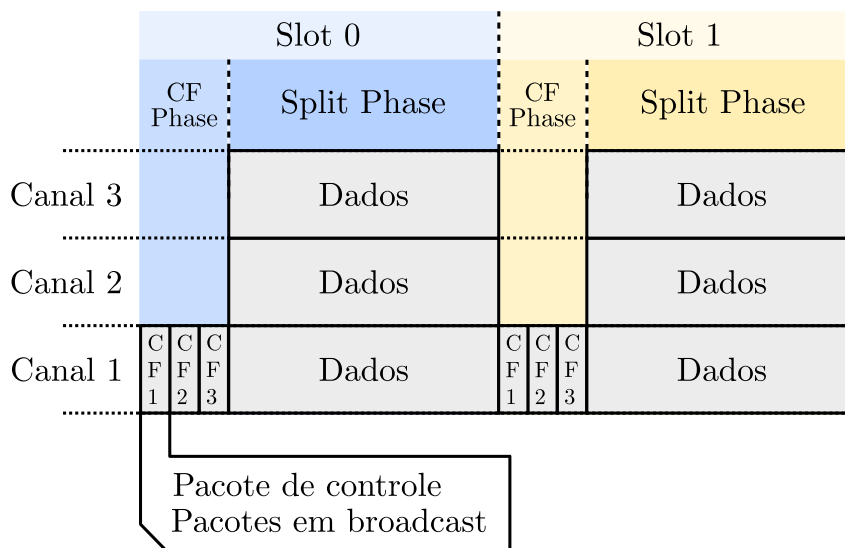
O *slot* selecionado por um nó é o único no qual ele pode transmitir dados. A Figura 4.10 ilustra dois *slots* em uma rede configurada para utilizar 3 canais. Durante a *CF Phase* desse *slot*, o nó transmite um pacote de controle em *broadcast*, informando o endereço do nó para o



qual enviará pacotes de dados na *Split Phase*. Esse pacote de controle é transmitido no *CF Slot* referente ao canal de operação do detentor do *slot*. Por exemplo, se um nó opera no canal 2, ele utilizará o *CF Slot 2* de seu *slot* para transmitir esse pacote de controle.

O nó notificado durante a *CF Phase* configura seu rádio para o canal do transmissor para receber os pacotes de dados. Como se observa na Figura 4.10, para essa rede com 3 canais podem ocorrer transmissões simultâneas pela rede em 3 canais diferentes na *Split Phase*. Com esse formato, o transmissor pode enviar pacotes a somente um nó de destino durante todo o *frame*. Os nós que não foram notificados no *CF Slot* desligam seus rádios na *Split Phase* para economizar energia.

Figura 4.10 – Formato dos *slots* do MC-LMAC.



Fonte: adaptado de (INCEL et al., 2011).

Aproveitando que todos os nós estarão configurados no canal comum, nessa implementação do MC-LMAC os nós também transmitem outros pacotes que tenham destino de *broadcast*, como pacotes de descoberta e de roteamento vindos da camada superior, durante os *CF Slots*. No entanto, como são *slots* com curta duração, a quantidade de pacotes transmitida foi limitada a 3 por *CF Slot*, sendo um pacote de controle e dois pacotes diversos.

Um nó pode ser selecionado como receptor por dois nós diferentes, que utilizam o mesmo *slot* mas em diferentes canais. Esse caso é chamado de disputa e o receptor sorteia de qual transmissor irá receber pacotes, e se configura para o canal desse transmissor. O outro transmissor deve tentar o envio novamente em seu próximo *slot*.

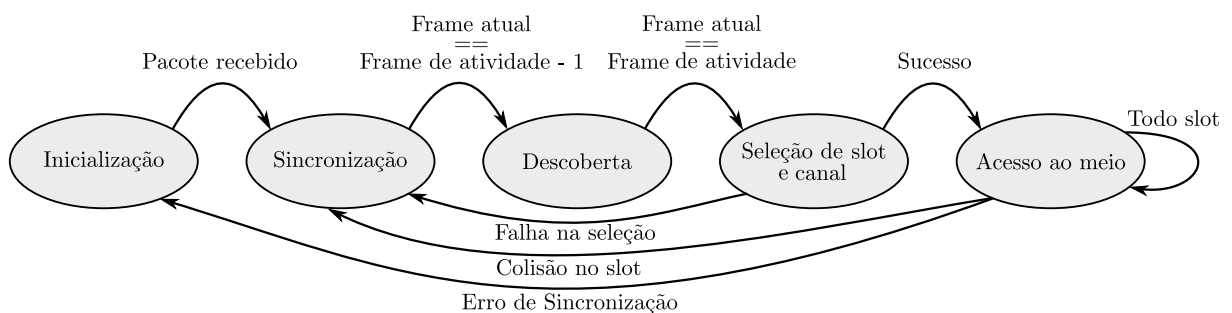
Na proposta original do MC-LMAC, para transmissões em *broadcast*, o nó de origem informa no *CF slot* o endereço de *broadcast* e todos os vizinhos se configuram para receber desse transmissor na *Split Phase*. Essas transmissões em *broadcast* podem aumentar a quantidade de disputas e o consumo de energia, dado que todos os vizinhos terão de ouvir essas transmissões.

A implementação do MC-LMAC feita neste trabalho modifica esse modo de operação, fazendo com que transmissões em *broadcast* só ocorram nos *CF Slots*, como especificado anteriormente, visando a reduzir a quantidade de disputas.

### Estados do MC-LMAC

A Figura 4.11 apresenta o diagrama de estados do MC-LMAC. Os nós iniciam no estado de *Inicialização*, ouvindo o canal comum em busca de pacotes de controle dos vizinhos. Ao detectar uma mensagem, o nó passa para o estado de *Sincronização* no qual irá sincronizar seu relógio interno com informações do pacote recebido. Na proposta original, o nó pode aguardar um ou mais *frames* para passar ao estado de descoberta, quando irá efetivamente coletar informações sobre o escalonamento da rede e tentar definir um par de *slot* e canal para sua operação.

Figura 4.11 – Diagrama de estados do MC-LMAC.



Fonte: adaptado de (INCEL et al., 2011).

Na implementação do MC-LMAC realizada para este trabalho, assim que o nó se sincroniza com a rede, ele passa imediatamente para o estado de *Descoberta*, no qual irá coletar informações sobre os nós vizinhos. Ao fim desse *frame*, o nó passa para o estado de *Seleção de slot e canal*, no qual irá determinar um *slot* e um canal de operação com base nas informações coletadas.

Se a seleção de *slot* e canal for bem sucedida, o nó passa para o estado de *Acesso ao meio*, no qual se mantém no restante de sua operação a menos que ocorra algum erro. Se no estado de

*Seleção de slot e canal* o nó não conseguir determinar um conjunto de operação, ele retorna ao estado de *Sincronização* e continua a tentar se configurar na rede. Do estado de *Acesso ao meio*, o nó pode retornar ao estado de *Inicialização* se detectar erros de sincronização. Essa detecção é feita pela comparação dos valores de *slot* atual e *frame* atual recebidos dos vizinhos.

### Pacote de controle

Na Figura 4.12 é apresentado o pacote de controle do MC-LMAC, cujo tamanho varia de acordo com a configuração do protocolo. O campo *ID* originalmente representa o endereço do nó e tem 16 bits, nessa implementação são utilizados os campos *Origem* e *Destino* para os pacotes, ficando o campo de *ID* para identificação individual de pacotes. No campo *Slots ocupados* é transmitido o vetor de ocupação dos nós vizinhos, para que nós a dois saltos tenham conhecimento da topologia.

Figura 4.12 – Pacote de controle do MC-LMAC.

ID	Origem		Destino		Slots ocupados	Meu Slot	Meu Canal	Slot
Frame	CFSlot	Hops	Slot de colisão	Canal de colisão	Confirmação	Receptor		

Fonte: adaptado de (INCEL et al., 2011).

Nos campos seguintes, *Meu Slot* e *Meu Canal*, o nó transmite qual seu conjunto de (slot, canal), enquanto nos campos *Slot*, *Frame* e *CFSlot* transmite os valores de *frame*, *slot* e *CF slot* atuais para sincronização. Os autores definem o campo *Hops* para auxílio na sincronização, porém não especificam um campo para transmissão do tempo global, o qual também foi omitido nessa implementação.

Os campos *Slot de colisão* e *Canal de colisão* são utilizados para informar a um nó que ocorreu colisão em seu *slot*, obrigando-o a tentar uma nova configuração. No campo *Receptor* o nó informa para qual vizinho irá transmitir dados durante a *Split Phase*.

No campo de *Confirmação* os nós colocam um bit 1 na posição de um transmissor do qual recebeu dados e 0 dos quais não recebeu, permitindo que um transmissor cuja transmissão não tenha sido confirmada faça uma nova tentativa de envio para esse nó. Com esse formato de confirmação o protocolo é incapaz de lidar com a perda de pacotes individuais dentro de um *slot*, pois é impossível determinar se um dos pacotes transmitidos durante a *Split Phase* foi perdido, caso vários tenham sendo transmitidos. Assim como no trabalho original do MC-LMAC, nos

testes realizados não foi utilizada confirmação e os pacotes foram enviados seguindo a política de melhor esforço (*best effort*), na qual os pacotes são transmitidos sem confirmação.

O tamanho do pacote de controle do MC-LMAC varia de acordo com a configuração do protocolo e podem ser maiores do que suportado por alguns transceptores. Supondo que uma rede foi configurada para ter 32 *slots* e 8 canais, os campos *Meu Slot*, *Slot* e *Slot de colisão* precisam ter pelo menos 5 bits, para que se possa representar os valores de *slots*. Os campos *Meu Canal*, *CFSlot* e *Canal de colisão* precisam ter pelo menos 3 bits, para representar os canais utilizados.

O maior consumo de espaço dos pacotes de controle vem dos campos de *Slots ocupados* e *Confirmação*, que têm tamanho igual a  $Quantidade\ de\ Slots \times Quantidade\ de\ Canais$ . No exemplo utilizado, cada um desses campos teria 256 bits ( $32\ slots \times 8\ canais$ ), que somados resultam em um tamanho superior ao suportado pelo rádio nRF24l01+, por exemplo. Com isso pode ser necessário fragmentar os pacotes de controle em diversos pacotes, aumentando o *overhead* de controle do protocolo.

## Roteamento

Para o roteamento dos pacotes com o MC-LMAC foi utilizado um protocolo já implementado no *framework* INET, chamado *WiseRoute*. O *WiseRoute* é um protocolo de roteamento que constrói uma árvore a partir de um ponto central, o *sink* da RSSF. O protocolo transmite pacotes em *broadcast* para descobrir a vizinhança e construir a árvore de roteamento.

Por utilizar roteamento em árvore, este protocolo é ideal para a comparação do MC-LMAC com o protocolo desenvolvido, dado que ambos utilizarão esquemas semelhantes para encaminhar os pacotes. No entanto, o *WiseRoute* suporta apenas tráfego convergente, ou seja, dos nós para o *sink*, enquanto o roteamento do Gen3 suporta tráfego bidirecional.

## 4.7 Considerações finais

Neste capítulo foi detalhada a metodologia de avaliação do protocolo Gen3. Inicialmente foram descritos os parâmetros de transmissão do protocolo, desde o tempo de duração dos *slots*, os tempos de guarda, até as configurações e características do rádio nRF24l01+, que foi utilizado como referência para as simulações.

Em seguida foram definidas as topologias, ou disposições dos nós, sendo elas divididas em topologias estáticas e mistas. Nas topologias estáticas, os nós permanecem na mesma po-

sição durante toda a simulação, enquanto nas topologias mistas alguns nós permanecem fixos cobrindo uma área por onde outros nós se movimentam.

Foram definidos dois tipos de tráfego, sendo eles: CBR, com geração constante de pacotes; e VBR, com geração de pacotes em rajadas. Cada tipo de tráfego permite a avaliação do protocolo quando submetido a diferentes condições da rede. Um formato de descrição dos cenários foi apresentado para guiar a análise dos resultados, compreendendo o nome do teste, os parâmetros utilizados, as topologias e finalmente as métricas avaliadas.

Por fim, é descrito o protocolo utilizado para comparação, o MC-LMAC, bem como algumas modificações realizadas em sua implementação. Essas modificações foram feitas visando à melhoria de alguns aspectos do protocolo, ou em virtude de falta de detalhes em sua descrição. No entanto, os principais conceitos do protocolo foram seguidos como descritos no trabalho original.

## 5 RESULTADOS E DISCUSSÕES

Neste capítulo são apresentados e discutidos os resultados obtidos com as simulações. Os resultados são subdivididos de acordo com os testes definidos na seção 4.5, sendo eles: *Configuração* (tempo de convergência da rede), *Transmissão de Dados*, *Mobilidade* e *Consumo Energético*. Em todas as simulações, e para cada combinação de parâmetros, foram executadas 30 repetições. Na análise dos resultados foi utilizado intervalo de confiança de 95%. Todos os gráficos apresentados possuem marcadores de intervalo de confiança, mas em vários casos os valores são mínimos e, portanto, imperceptíveis nos gráficos.

### 5.1 Tempo de convergência da rede

Nos testes de configuração foi avaliado o tempo médio necessário para a convergência da rede, dado pelo tempo de associação de todos os nós, e posteriormente pelo tempo necessário para que a rede atinja sua configuração final, denominado tempo de estabilização. Considera-se que a rede se encontra estável quando o ciclo já possui a quantidade de *frames* igual à quantidade de nós, e todos os nós já possuem a quantidade necessária de *frames*, dada pelo número de nós em sua sub-rede mais um.

Nesses testes foi analisado o impacto do uso de CCC na configuração da rede, o impacto de diferentes valores do parâmetro *Settle Cycles* e da quantidade de canais. A Tabela 5.1 traz os parâmetros adicionais utilizados nos testes de configuração.

Tabela 5.1 – Parâmetros complementares do Gen3 para os testes de configuração.

Parâmetro	Valor
<i>Settle Cycles</i>	0, 1, 2, 4, 8, 12, 16, 20
Quantidade de canais	2, 4, 6, 8, 10

Fonte: do autor (2017).

#### 5.1.1 Tempo de associação

Os resultados avaliados nessa seção são destacados na Figura 5.1.

Em todos esses testes a rede foi iniciada tendo apenas o *sink* associado e transmitindo pacotes de controle C1 para a associação de novos nós. Assim que um nó se associa, ele já pode transmitir dados pela rede, mas um *frame* deverá ser adicionado à rede, gerando uma nova configuração. Enquanto essa configuração não é gerada, os nós operam normalmente, mas o nó pai fica sem *frames* livres para permitir a associação de novos nós. O filho recém-associado

Figura 5.1 – Cenário de configuração avaliado: associação.

$$\text{Configuração - S/MXC} \left[ \begin{array}{l} \text{CCC} \\ \text{S/CCC} \end{array} \right] \left[ \begin{array}{cc} 25\text{D}25 & 25\text{D}75 \\ 25\text{D}100 & 25\text{D}150 \\ \text{L12} & \text{L23} \end{array} \right] \left[ \begin{array}{l} \text{Tempo de Associação} \\ \text{Tempo de Estabilização} \end{array} \right]$$

Fonte: do autor (2017).

também irá aguardar um período (baseado em *Settle Cycles*, descrito na Seção 3.2) para oferecer associação a novos nós.

Além desse parâmetro, o Gen3 pode utilizar um canal de controle comum (CCC) ou não. Caso use, basta que o nó entrando na rede ouça as transmissões no CCC para tentar se associar. Caso não use CCC, o nó deve varrer todas as frequências da lista de canais buscando por um pacote de associação.

Os resultados apresentados nessa seção servem para analisar os impactos de três parâmetros no tempo de associação dos nós, sendo eles: valor de *Settle Cycles*, CCC ativado ou desativado e quantidade de canais. O tempo de associação marca apenas a última vez em que o nó se conectou na rede, como no caso de um nó que se associou, mas devido a interferências se desassociou e tentou uma nova associação em outro ponto. De cada uma das 30 repetições foi calculado o tempo médio de associação dos nós. Posteriormente foi calculada a média desses 30 valores para se chegar ao resultado apresentado nos gráficos.

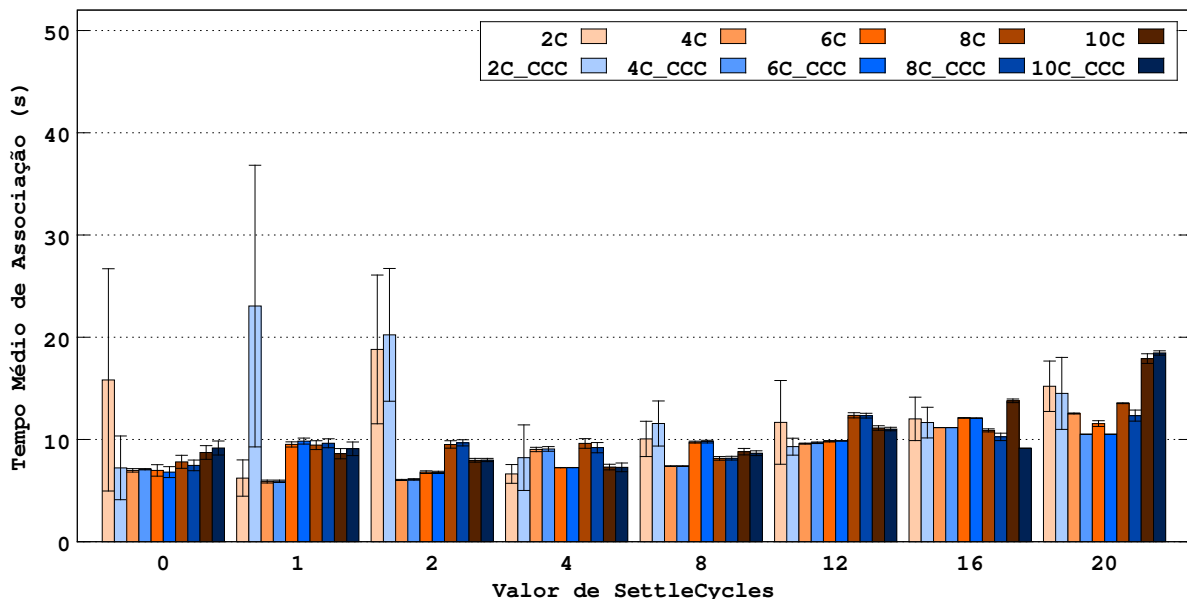
A legenda dos gráficos descreve quantos canais foram utilizados e se a configuração utilizava CCC ou não. Por exemplo, a legenda 6C se refere aos testes com 6 canais e sem uso de CCC, enquanto 6C\_CCC indica os testes com 6 canais com o uso de CCC.

O gráfico da Figura 5.2 apresenta o tempo médio de associação dos nós na topologia 25D25. Nessa topologia, o uso de CCC não resultou em redução do tempo na maioria dos casos, pois com a maior densidade da rede, a probabilidade de um nó encontrar um pacote de associação sendo transmitido por algum vizinho é maior.

As configurações com poucos canais e valor mais baixo de *Settle Cycles* apresentaram as maiores variações, com tempos mais altos de associação. Nesses casos, um nó recém-associado aguarda menos tempo para transmitir pacotes para associação de novos nós, fazendo com que a rede cresça mais rapidamente em altura. Aliado a isso, com o uso de poucos canais ocorre o reuso de um conjunto (*slot*, canal) a menos saltos de distância, o que pode causar colisões a gerar a desconexão dos nós.

Após a desconexão, os nós podem se reconectar a outros que estejam a menos saltos do *sink*, e portanto, evitar a reutilização de *slots* e canais. Esse problema pode atrasar a associação de alguns nós mas não impede a entrada deles na rede e pode ser resolvido definindo uma quantidade maior de canais, como observado nos resultados com 4 canais mostrando que a adição de apenas dois canais já resultou em uma operação mais estável.

Figura 5.2 – Tempo médio de associação na topologia 25D25.



Fonte: do autor (2017).

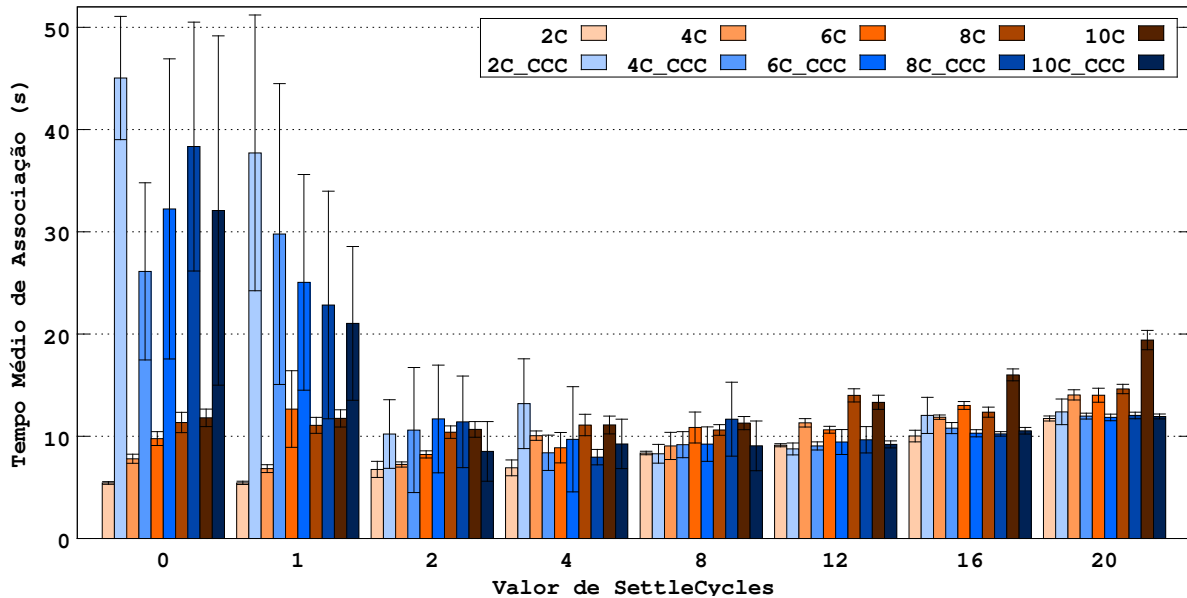
No gráfico da Figura 5.3 são mostrados os valores obtidos na topologia 25D75, que tem densidade menor que a 25D25. Os valores obtidos com *Settle Cycles* iguais a 0 e 1 apresentaram comportamento irregular, com grandes oscilações utilizando CCC. A combinação de dois fatores é a provável causa desse comportamento nesses casos:

- O rápido crescimento da rede em altura quando o valor de *Settle Cycles* é mais baixo, o que força a reutilização de conjuntos de *slot* e canal em distâncias menores.
- A distância de interferência do sinal. Ainda que a determinada distância um nó não consiga decodificar o sinal recebido e traduzí-lo em informação, esse sinal pode atrapalhar outras transmissões para ele. Nessa topologia a quantidade de nós no raio de alcance é menor, mas vários outros se encontram a distâncias suficientes para que suas transmissões causem ruídos, interferindo na recepção de pacotes.



Os demais resultados apresentaram valores mais regulares e tempos significativamente menores. O impacto de uso ou não de CCC nos demais casos também foi menor em relação aos demais testes.

Figura 5.3 – Tempo médio de associação na topologia 25D75.



Fonte: do autor (2017).

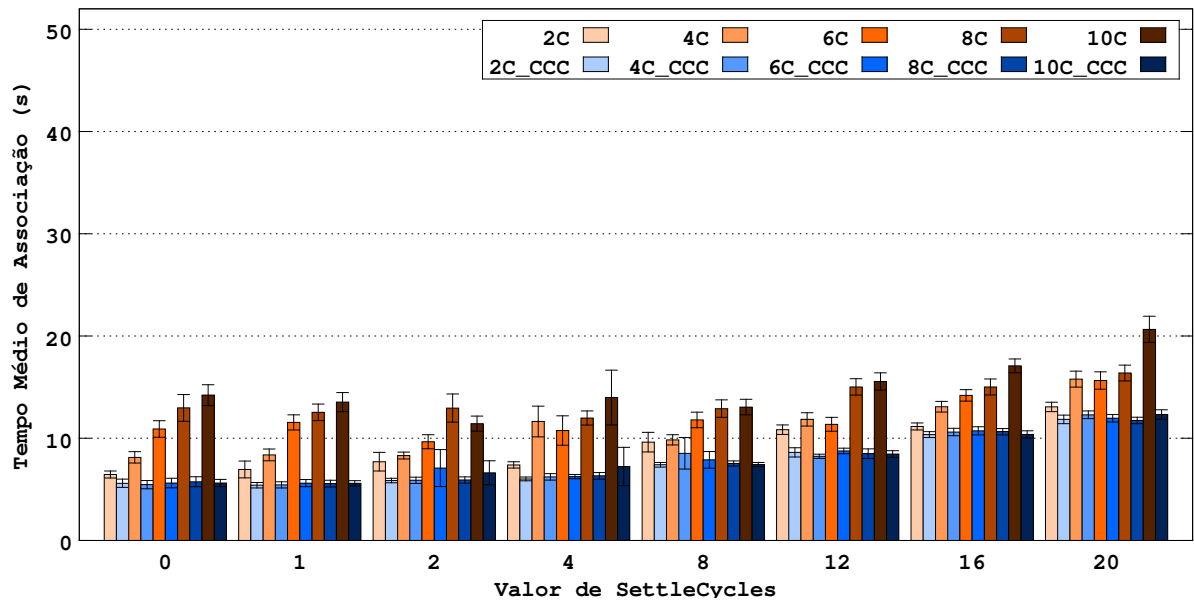
Os gráficos das Figuras 5.4 e 5.5 trazem os resultados obtidos com as topologias 25D100 e 25D150. A topologia 25D150 apresenta tempos sutilmente mais altos que a topologia 25D100 por ser mais esparsa, mas em ambas o comportamento do Gen3 foi semelhante.

As configurações utilizando CCC apresentaram resultados similares para um mesmo valor de *Settle Cycles*, enquanto que configurações sem CCC tendem a apresentar tempos mais elevados com maiores quantidades de canais. Esse comportamento é esperado, pois com mais canais os nós precisam buscar os pacotes de associação em várias frequências diferentes, atrasando a associação.

No gráfico da Figura 5.6 são mostrados os tempos médios de associação na topologia L12. Nessa topologia fica mais evidente o impacto do valor de *Settle Cycles* no tempo médio de associação dos nós. Por ser uma topologia linear, o último nó só irá se associar após decorrida a quantidade de ciclos necessárias para que o nó anterior comece a transmitir pacotes de associação. O uso de CCC não causou impacto significativo na vasta maioria dos casos.

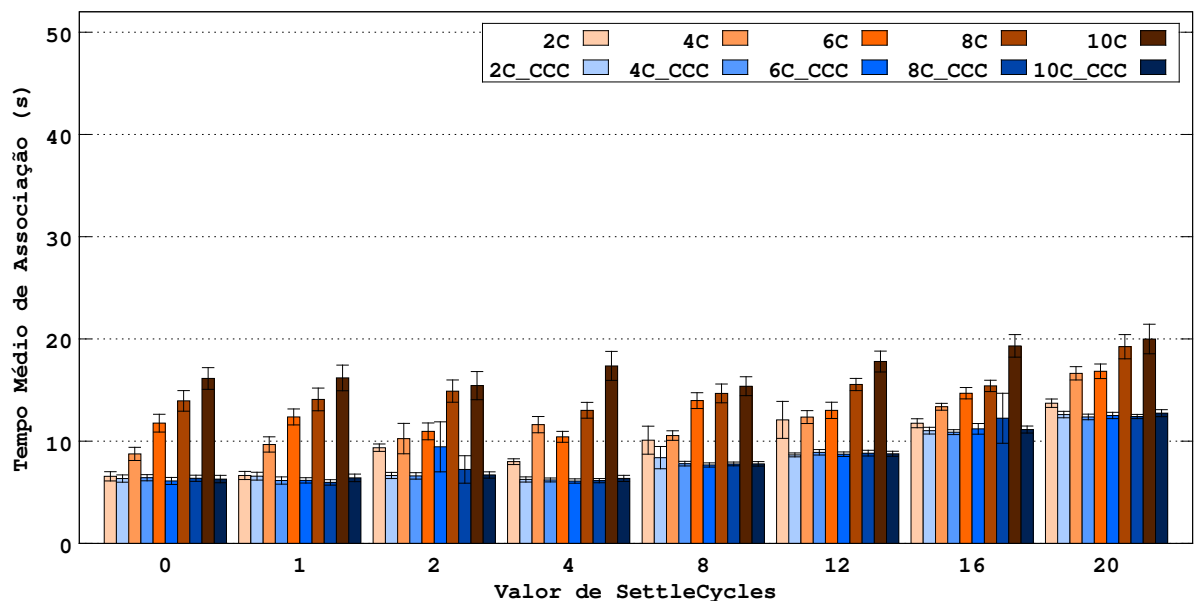
O gráfico da Figura 5.7 traz os resultados obtidos para a topologia L23, também linear mas com o *sink* em posição central e com mais nós. O uso de CCC também não resultou em

Figura 5.4 – Tempo médio de associação na topologia 25D100.



Fonte: do autor (2017).

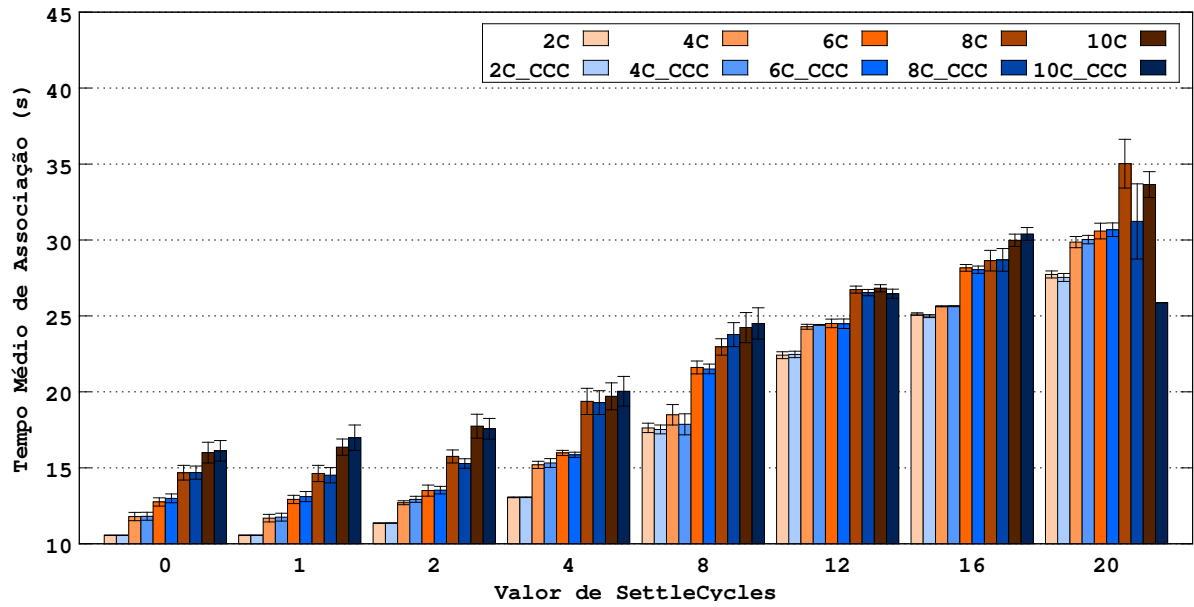
Figura 5.5 – Tempo médio de associação na topologia 25D150.



Fonte: do autor (2017).

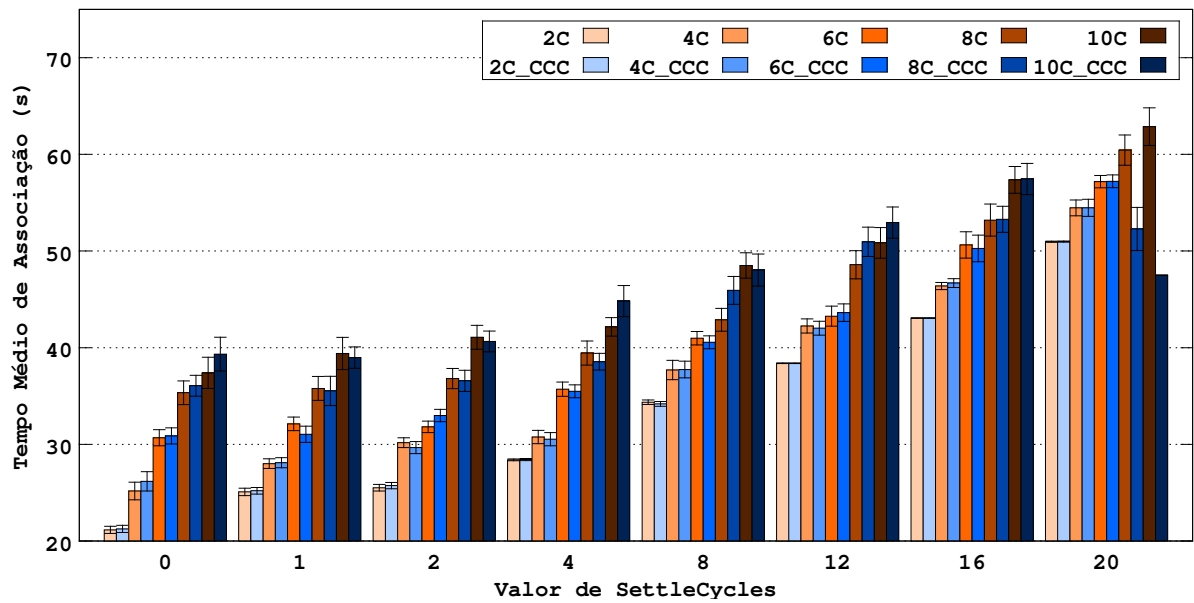
vantagens na maioria dos casos e os valores obtidos foram similares aos da topologia com 12 nós.

Figura 5.6 – Tempo médio de associação na topologia L12.



Fonte: do autor (2017).

Figura 5.7 – Tempo médio de associação na topologia L23.



Fonte: do autor (2017).

### 5.1.2 Tempo de estabilização da rede

Os resultados avaliados nessa seção são destacados na Figura 5.8.

Figura 5.8 – Cenário de configuração avaliado: estabilização.

$$\text{Configuração - S/MXC} \begin{bmatrix} \text{CCC} \\ \text{S/CCC} \end{bmatrix} \begin{bmatrix} 25\text{D}25 & 25\text{D}75 \\ 25\text{D}100 & 25\text{D}150 \\ \text{L12} & \text{L23} \end{bmatrix} \begin{bmatrix} \text{Tempo de Associação} \\ \text{Tempo de Estabilização} \end{bmatrix}$$

Fonte: do autor (2017).

Com a entrada de novos nós na rede, mais *frames* devem ser adicionados ao ciclo. Para isso, o *sink* gera uma nova configuração com a quantidade correta de *frames*, e estes são redistribuídos pela rede de acordo com as necessidades dos nós. Quando uma nova configuração é gerada o *sink* define o ciclo de *deadline*, no qual essa nova configuração passará a ser utilizada por todos os nós. O tempo de estabilização marca o momento em que a configuração final da rede passou a ser utilizada, com todos os *frames* necessários no ciclo e com a alocação da quantidade correta para cada nó, de acordo com as definições do protocolo.

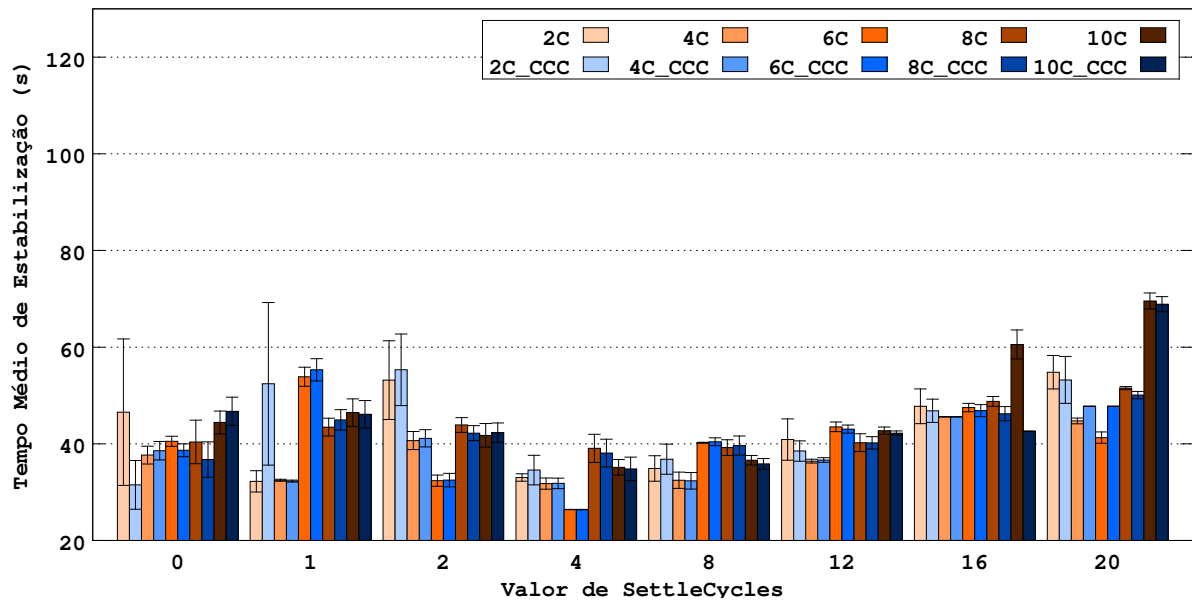
A estabilização da rede é diretamente relacionada ao tempo de associação. O protocolo não conhece previamente a quantidade de nós existentes, mas se adapta à chegada ou descoberta de novos nós. Assim, quanto mais tempo decorrer até a associação de todos os nós em uma área, mais tempo essa rede levará para chegar à sua configuração final. Como dito anteriormente, o fato da rede não ter se estabilizado não impede que os nós já associados transmitam pacotes para qualquer outro nó. Entretanto, isso pode limitar a entrada de novos nós, pois a quantidade de *frames* livres é menor.

A Figura 5.9 traz o gráfico com os tempos de estabilização na topologia 25D25. O comportamento observado tem clara relação com o tempo de associação, com tempos mais elevados em configurações sem CCC. São observadas maiores variações principalmente com o uso de valores menores para *Settle Cycles* e poucos canais.

No gráfico da Figura 5.10, para a topologia 25D75, são observadas maiores oscilações para valores de *Settle Cycles* iguais a 0 e 1. Na maior parte dos demais casos foram observadas variações desprezíveis no tempo de estabilização com e sem o uso de CCC.

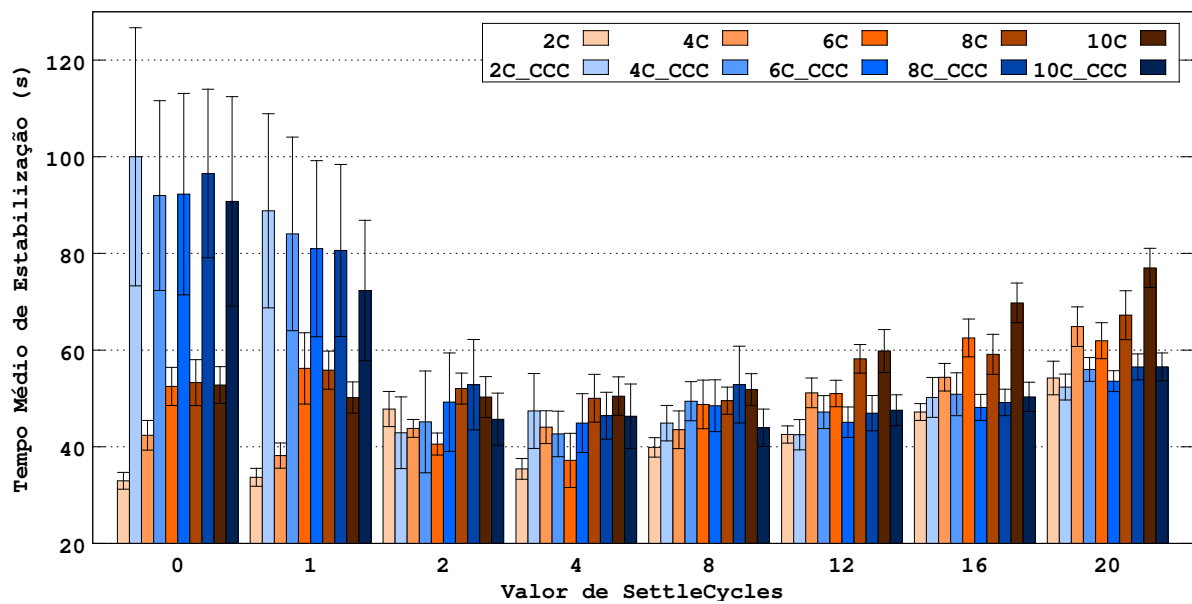
Nas topologias 25D100 e 25D150, cujos resultados são apresentados nos gráficos das Figuras 5.11 e 5.12 foram observados comportamentos estáveis e consonantes com os valores de tempo de associação. Em geral, a rede se estabilizou mais rápido nas configurações com CCC, dado que os nós também se associam mais rapidamente.

Figura 5.9 – Tempo médio de estabilização na topologia 25D25.



Fonte: do autor (2017).

Figura 5.10 – Tempo médio de estabilização na topologia 25D75.

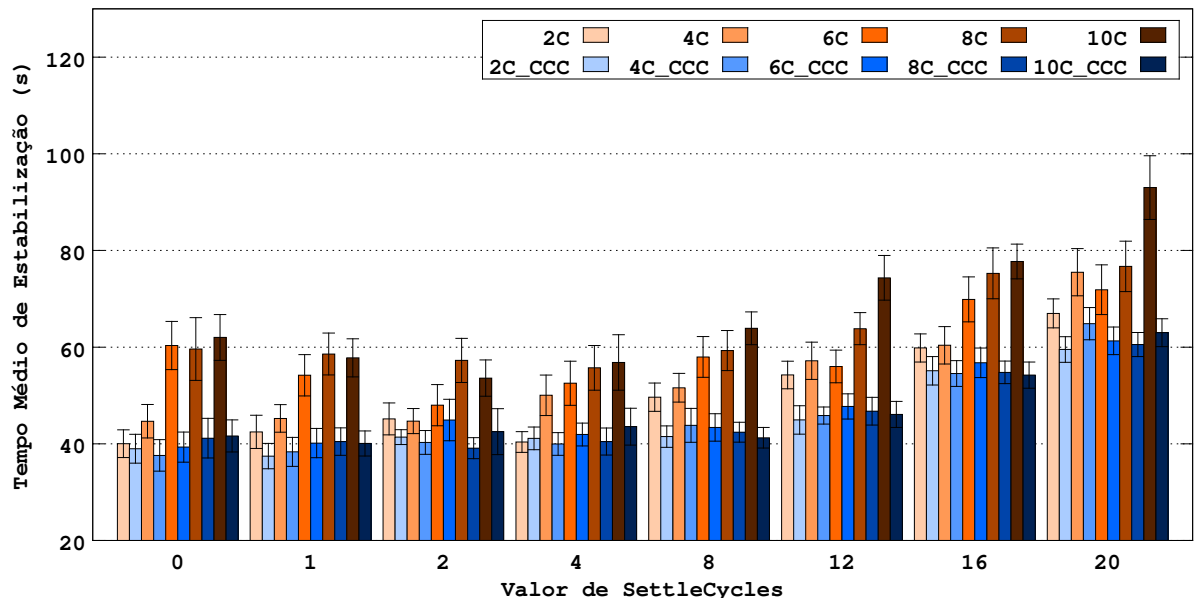


Fonte: do autor (2017).

Para as topologias lineares L12 e L23, os tempos médios de estabilização são apresentados nos gráficos das Figuras 5.13 e 5.14. Da mesma forma que nos resultados anteriores, o tempo de estabilização teve relação direta com o tempo de associação médio dos nós.

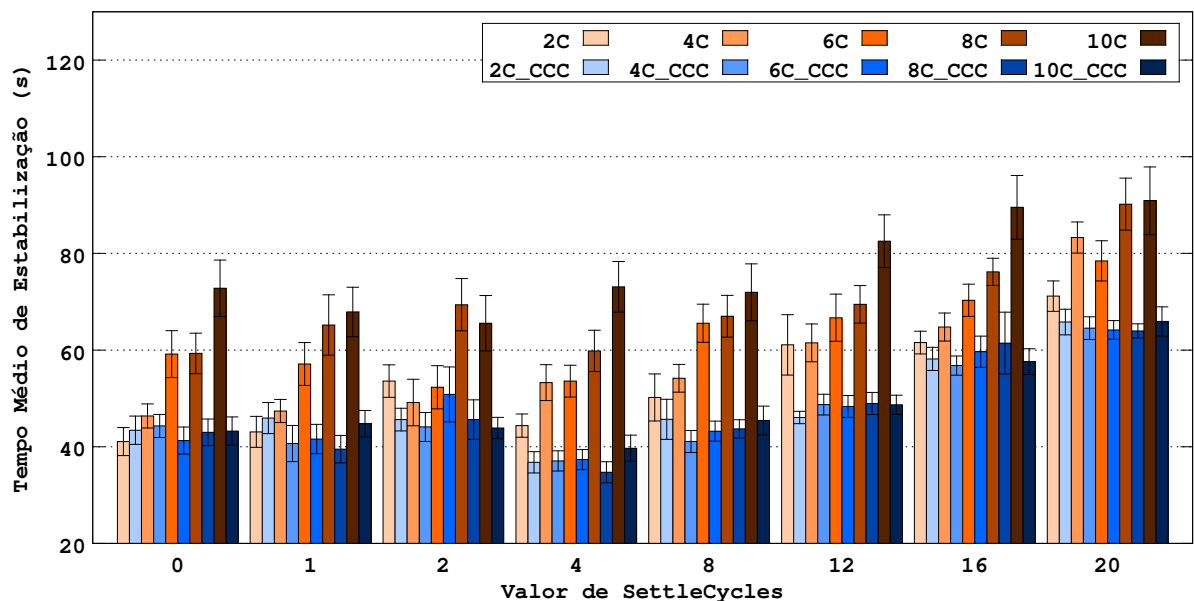
No pior caso da topologia L12, a rede chegou à configuração final em tempo inferior a 120 segundos, na configuração de 8 canais sem CCC. Nessa mesma configuração, o tempo

Figura 5.11 – Tempo médio de estabilização na topologia 25D100.



Fonte: do autor (2017).

Figura 5.12 – Tempo médio de estabilização na topologia 25D150.

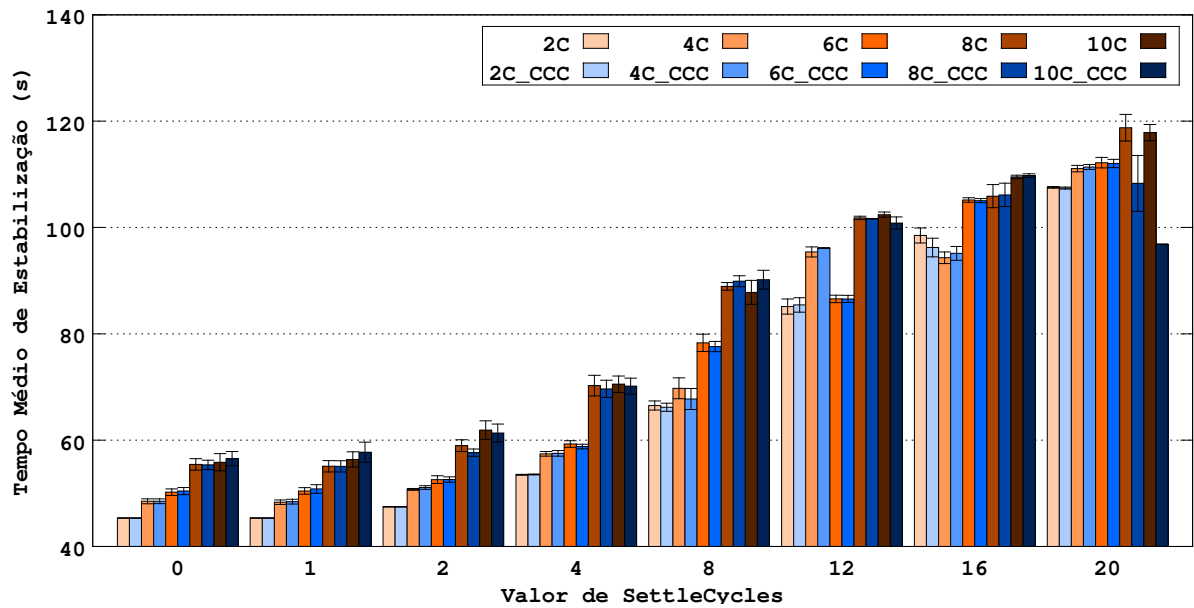


Fonte: do autor (2017).

médio de associação dos nós foi de 35 segundos. No melhor caso, com 2 canais o tempo de estabilização médio foi de aproximadamente 45 segundos, com *Settle Cycles* igual a 0 ou 1, enquanto os nós apresentaram tempo médio de associação de 10 segundos.

Na topologia L23, o pior caso ocorreu com *Settle Cycles* igual a 20 e 10 canais sem CCC, com tempo próximo a 245 segundos. Enquanto o melhor caso ocorreu para *Settle Cycles* igual

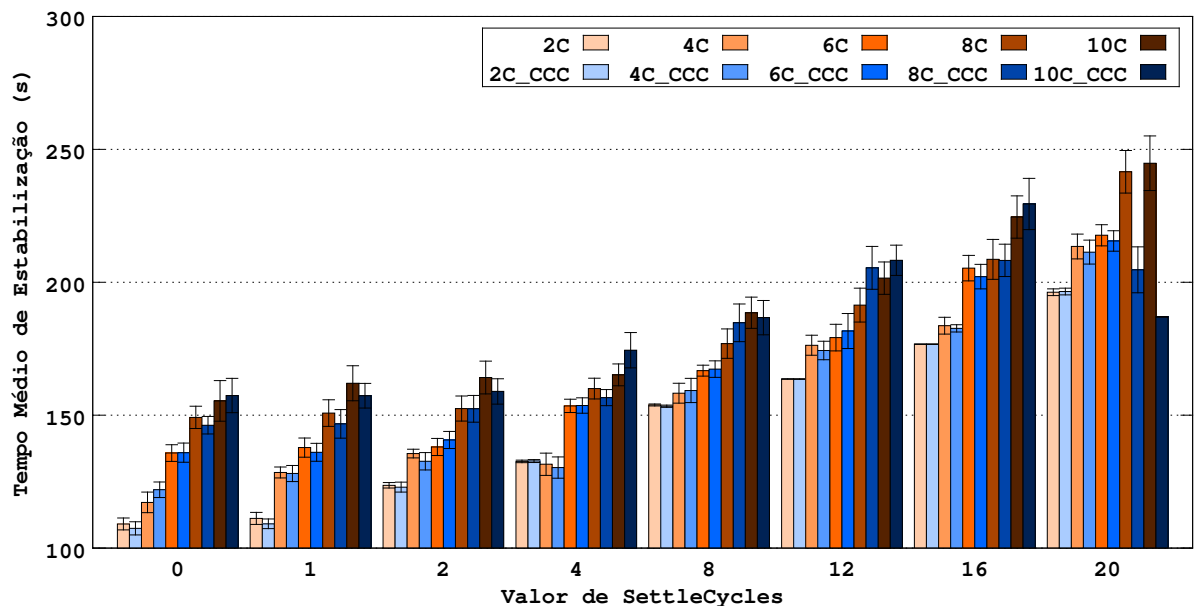
Figura 5.13 – Tempo médio de estabilização na topologia L12.



Fonte: do autor (2017).

a 0, e 2 canais, com tempo próximo a 110 segundos. Como as topologias lineares são esparsas, o uso de mais canais teve impacto negativo no tempo de associação, dado que sempre que um conjunto (*slot*, canal) for reutilizado, os nós alocados ao mesmo conjunto estarão fisicamente distantes o suficiente para não causarem interferências.

Figura 5.14 – Tempo médio de estabilização na topologia L23.



Fonte: do autor (2017).

### 5.1.3 Considerações finais

Os resultados apresentados nesta seção indicam uma relação direta entre os tempos médios de associação dos nós e os tempos de estabilização. Esse comportamento é o esperado para o protocolo, pois ele se adapta à medida que novos nós entram na região da rede ou são descobertos.

Também foram avaliados os impactos da quantidade de canais e dos parâmetros do protocolo na convergência das redes. O uso de CCC permitiu a redução do tempo de convergência na maioria dos casos. Por outro lado, sem o uso de CCC é possível observar um crescimento do tempo de convergência à medida que a quantidade de canais é incrementada, também sendo um comportamento esperado, dada a necessidade de que os nós verifiquem vários canais buscando pacotes de associação.

Os valores do parâmetro *Settle Cycles* também tiveram influência na convergência da rede. Valores menores desse parâmetro resultaram em menor tempo de convergência nas redes de menor densidade. Contudo, dependendo da densidade das redes, como na topologia 25D25, baixos valores de *Settle Cycles* (0, 1 e 2) combinados ao uso de poucos canais (2 canais) resultaram em aumento do tempo de convergência. Outro caso foi na topologia 25D75, que com valores de 0 e 1 para *Settle Cycles*, e CCC ativado resultou em atrasos consideráveis na associação de nós e estabilização da rede.

Pela análise dos resultados, a configuração do Gen3 com valor de *Settle Cycles* igual a 4, 6 canais e CCC ativado apresentou resultados satisfatórios na maioria dos testes realizados.



## 5.2 Transmissão de dados

Os testes com transmissão de dados avaliaram a capacidade de entrega de pacotes dos protocolos. Inicialmente são analisados os resultados com tráfego CBR, com geração constante de pacotes a diferentes intervalos e com todos os pacotes destinados ao nó *sink*. Posteriormente são analisados os testes com tráfego VBR, nos quais são transmitidos pacotes em rajadas com padrões variados e destinados a diferentes nós da rede.

A Tabela 5.2 traz os parâmetros adicionais do Gen3 para os testes de transmissão. Com base nos resultados de configuração, *Settle Cycles* foi fixado em 4, a quantidade de canais em 6 e o limiar de 1% de ocupação de *buffer* para entrar em MXC. Nos gráficos apresentados, Gen3 se refere ao protocolo desenvolvido com multiplexação desativada, enquanto Gen3S referencia o protocolo com a MXC ativada.

Tabela 5.2 – Parâmetros complementares do Gen3 para os testes de transmissão.

Parâmetro	Valor
<i>Settle Cycles</i>	4
Quantidade de canais	6
Uso de CCC	Sim
Limiar de MXC (quando ativada)	1%

Fonte: do autor (2017).

A Tabela 5.3 traz os parâmetros do MC-LMAC para cada topologia, dado que suas configurações variam com a densidade da rede. Para cada teste executado com o MC-LMAC foram testadas diferentes quantidades de *slots* e canais. Foi utilizada a quantidade mínima de *slots* que permitisse a configuração da rede de acordo com a densidade de nós. A quantidade de *slots* e canais utilizada em cada teste foi a que permitiu que o protocolo apresentasse os melhores resultados.

Tabela 5.3 – Parâmetros do MC-LMAC para os testes de transmissão.

Parâmetro	Topologia	Valor
Quantidade de <i>slots</i>	25D25	25
	25D150	4
	L12	3
	L23	3
Quantidade de canais	25D25	6
	25D150	4
	L12	2
	L23	2
Duração do <i>slot</i>	-	40 ms

Fonte: do autor (2017).

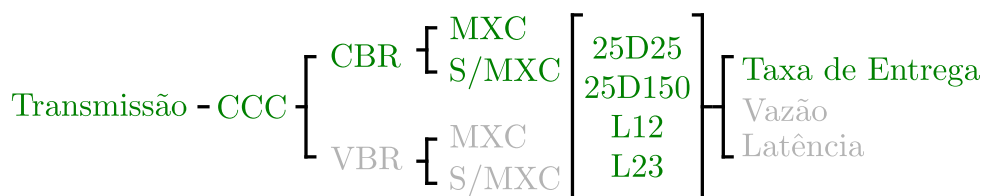
Os resultados dos testes preliminares para a definição de parâmetros do MC-LMAC não foram incluídos nessa seção por estarem fora do escopo do trabalho. Os valores apresentados para este protocolo são os melhores obtidos com os testes.

Cada simulação teve duração de 300 segundos, com início da transmissão de dados aos 70 segundos e encerramento aos 270 segundos. O período de 30 segundos entre o fim da transmissão e o fim da simulação visa dar oportunidade para que todos os nós descarreguem seus *buffers*, permitindo a análise correta da taxa de entrega.

### 5.2.1 Taxa de entrega - CBR

Os resultados avaliados nessa seção são destacados na Figura 5.15. Nesse teste todos os nós, exceto o *sink*, geravam pacotes destinados ao *sink*. Os pacotes eram gerados pela camada de aplicação e recebidos pelo protocolo MAC, que os colocavam em *buffer*. Caso o *buffer* estivesse cheio, os pacotes mais antigos eram descartados para dar lugar aos novos recebidos, e esse descarte contabilizado. A soma dos pacotes gerados por todos os nós foi contabilizada como a quantidade de pacotes criados.

Figura 5.15 – Cenário de transmissão avaliado: taxa de entrega com tráfego CBR.

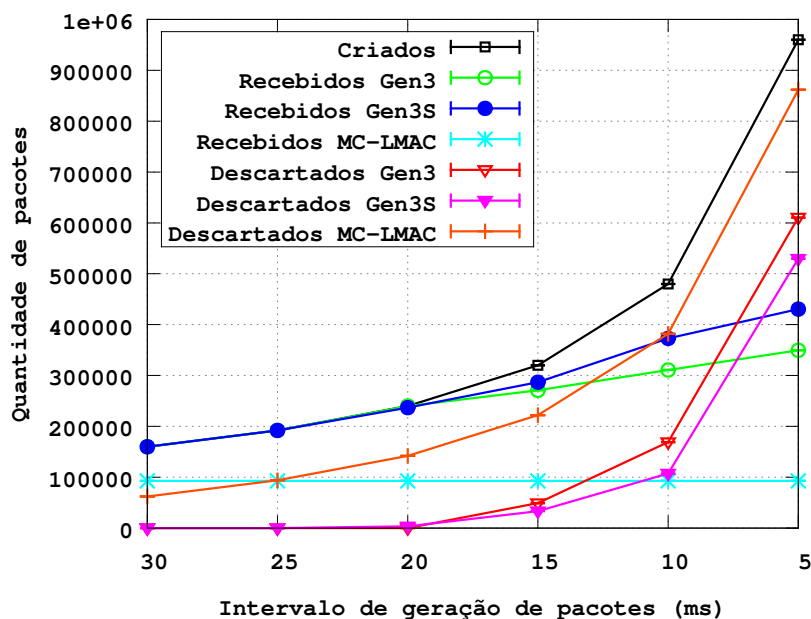


Fonte: do autor (2017).

A implementação do MC-LMAC utilizada não identifica perdas de pacotes e não possui mecanismo de retransmissão. Já o Gen3 utiliza retransmissões e contabiliza uma perda de pacote quando todas as tentativas de retransmissão falham. A perda de pacotes no Gen3, quando ocorrida, apresentou valores médios inferiores a 0,001%. Por esse motivo, os dados de pacotes perdidos não são exibidos nos gráficos.

O gráfico da Figura 5.16 apresenta os resultados de entrega de pacotes na topologia 25D25. Os resultados mostram larga vantagem do Gen3 sobre o MC-LMAC em todos os intervalos de geração de pacotes. Com intervalo de 20 ms entre pacotes, resultando em 50 pacotes por segundo gerados por cada nó, o Gen3 foi capaz de atingir taxas de entrega próximas ou iguais a 100%.

Figura 5.16 – Entrega de pacotes na topologia 25D25 com tráfego CBR.



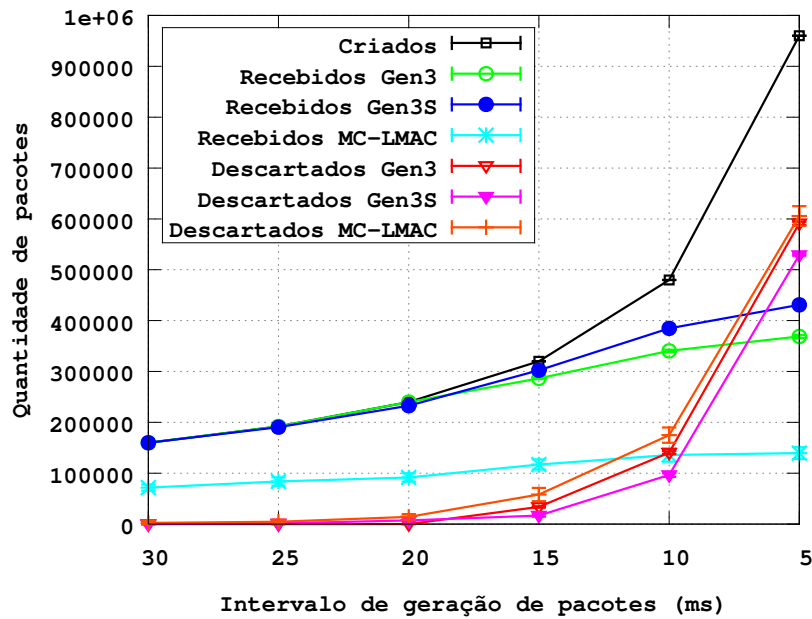
Com intervalos entre pacotes inferiores a 20 ms, resultando em 66, 100 e 200 pacotes por segundo gerados por cada nó, cresce o descarte de pacotes por escassez de *buffer*. O uso de MXC permitiu que o protocolo reduzisse o descarte e aumentasse a entrega de pacotes com o crescimento do tráfego, atingindo taxa de entrega acima de 400% superior ao MC-LMAC no tráfego mais intenso.

No gráfico da Figura 5.17 são apresentados os resultados para uma topologia mais esparsa. Por ser uma rede mais esparsa, o MC-LMAC pode ser configurado com menos *slots*, permitindo um *frame* menor, e maior reutilização de *slots* em canais diferentes.

O Gen3 continua necessitando dos 25 *frames* por ciclo, independente da densidade, resultando em um ciclo com duração de 1 segundo. O ciclo (chamado de *frame* no MC-LMAC) nessa configuração teve 160 ms, sendo consideravelmente menor. Isso, no entanto, não resultou em ganho de performance significativo do MC-LMAC, sendo superado pelo Gen3 em todos os padrões de tráfego testados. Assim como no caso anterior, o uso de MXC permitiu aumentar a quantidade de pacotes entregues e reduzir o descarte à medida que o tráfego na rede foi intensificado.

A Figura 5.18 traz o gráfico de entrega de pacotes para a topologia L12, que tem apenas 12 nós dispostos de forma linear. Com menos nós também são gerados menos pacotes e as-

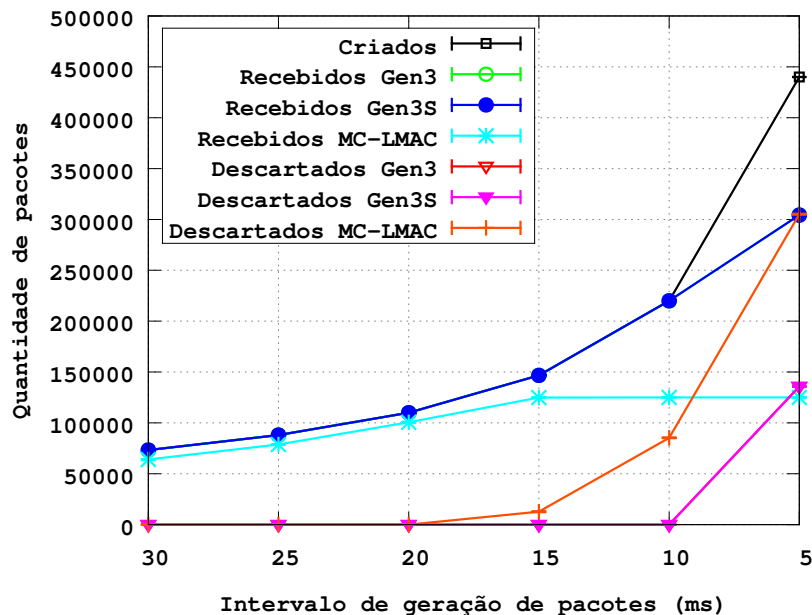
Figura 5.17 – Entrega de pacotes na topologia 25D150 com tráfego CBR.



Fonte: do autor (2017).

sim, observa-se um comportamento próximo do Gen3 e o MC-LMAC até a configuração com intervalo de 15 ms entre pacotes.

Figura 5.18 – Entrega de pacotes na topologia L12 com tráfego CBR.



Fonte: do autor (2017).

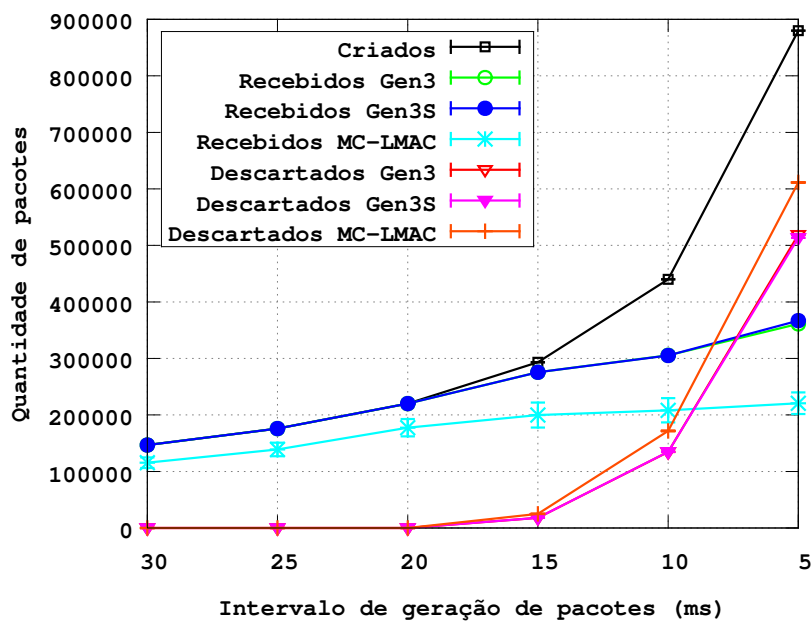
A partir desse valor, o MC-LMAC chega ao limite da capacidade e continua entregando uma quantidade constante de pacotes nos demais padrões de tráfego. O Gen3 é capaz de entregar

100% dos pacotes nessa topologia até o intervalo de 10 ms, entregando mais que o dobro de pacotes que o MC-LMAC.

Nessa topologia, o uso de MXC não influenciou nos resultados e os valores para o Gen3 foram sobrepostos pelos resultados do Gen3S no gráfico. Da forma como a MXC foi aplicada ao protocolo, só é possível obter alguma vantagem caso um nó tenha mais que um filho. Como cada nó tem apenas um filho, o restante do *slot* não pode ser aproveitado por nenhum outro nó, cabendo ao pai e ao filho desativarem seus rádios para economizar energia.

Na topologia L23, cujo gráfico é apresentado na Figura 5.19, o Gen3 também supera o MC-LMAC com todos os padrões de tráfego. Com intervalos inferiores a 20 ms já se observa descarte de pacotes tanto no Gen3 quanto no MC-LMAC. A vantagem obtida com MXC nessa topologia é desprezível. O *sink*, que é posicionado no centro da rede, é o único nó a ter mais de um filho. Dessa forma, apenas os dois filhos do *sink* entram efetivamente em MXC.

Figura 5.19 – Entrega de pacotes na topologia L23 com tráfego CBR.



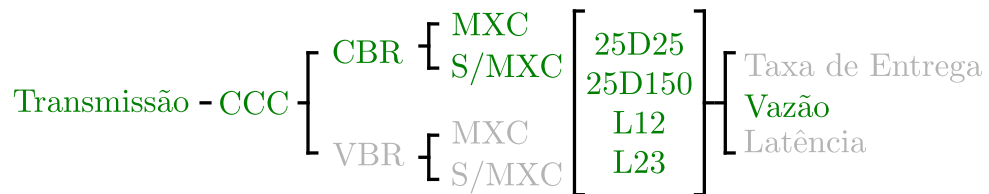
Fonte: do autor (2017).

No MC-LMAC foi possível manter a mesma configuração para a topologia L12, tendo apenas 3 *slots* por *frame* e 2 canais, resultando em um *frame* de 120 ms. Apesar de ser a configuração com melhor resultado para o MC-LMAC, ela não se traduziu em vantagem sobre o Gen3.

### 5.2.2 Vazão da rede - CBR

Os resultados avaliados nessa seção são destacados na Figura 5.20. Para medir a vazão foram contabilizados quantos pacotes chegavam ao *sink* por segundo. Como a geração de pacotes é constante, essa quantidade tem relação direta com a capacidade de entrega dos nós.

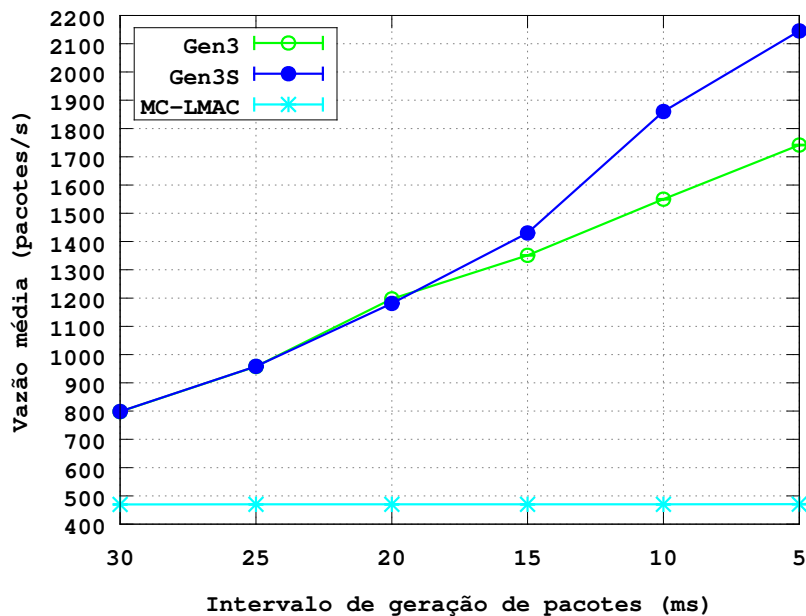
Figura 5.20 – Cenário de transmissão avaliado: vazão com tráfego CBR.



Fonte: do autor (2017).

O gráfico da Figura 5.21 traz os resultados obtidos na topologia 25D25. Enquanto o MC-LMAC entrega uma quantidade fixa inferior a 500 pacotes por segundo, o Gen3 apresenta um crescimento constante em sua configuração normal, acompanhando a quantidade de tráfego gerada, sendo capaz de oferecer vazão ainda maior com o uso de MXC.

Figura 5.21 – Vazão média de pacotes na topologia 25D25 com tráfego CBR.

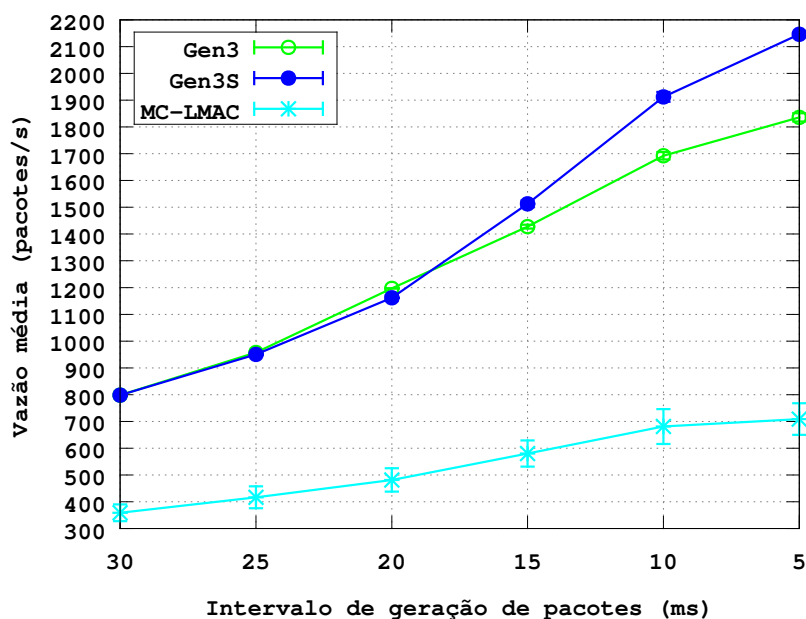


Fonte: do autor (2017).

Com intervalos entre pacotes inferiores a 20 ms é possível observar o crescimento da vazão com o uso de MXC, como previsto, atingindo taxas de até 500 pacotes por segundo a mais que sua configuração sem MXC, para o padrão com 5 ms de intervalo.

A configuração do MC-LMAC para a topologia mais esparsa permitiu o aumento considerável da sua capacidade de vazão. O valor obtido passou de uma quantidade constante inferior a 500 pacotes por segundo na topologia 25D25, para cerca de 700 pacotes por segundo na topologia 25D150, cujos resultados apresentados no gráfico da Figura 5.22.

Figura 5.22 – Vazão média de pacotes na topologia 25D150 com tráfego CBR.



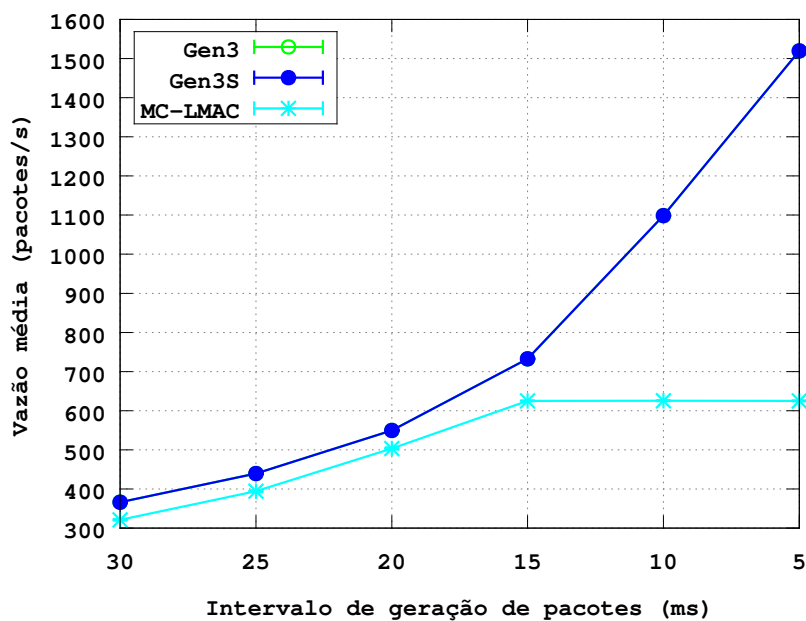
Fonte: do autor (2017).

No entanto, o Gen3 ainda supera os valores de vazão obtidos em todos os casos, atingindo vazão superior em alguns casos, mesmo em sua configuração sem MXC. Como observado na topologia anterior, com MXC foi possível obter maior vazão com a intensificação do tráfego.

Consonante com os resultados de entrega para a topologia L12, o MC-LMAC teve sua capacidade de vazão saturada a partir do intervalo de 15 ms entre pacotes, como mostrado no gráfico da Figura 5.23. A vazão obtida com o Gen3 cresceu com o aumento do tráfego, sem que fossem observadas vantagens com o uso de MXC, pelos motivos discutidos anteriormente.

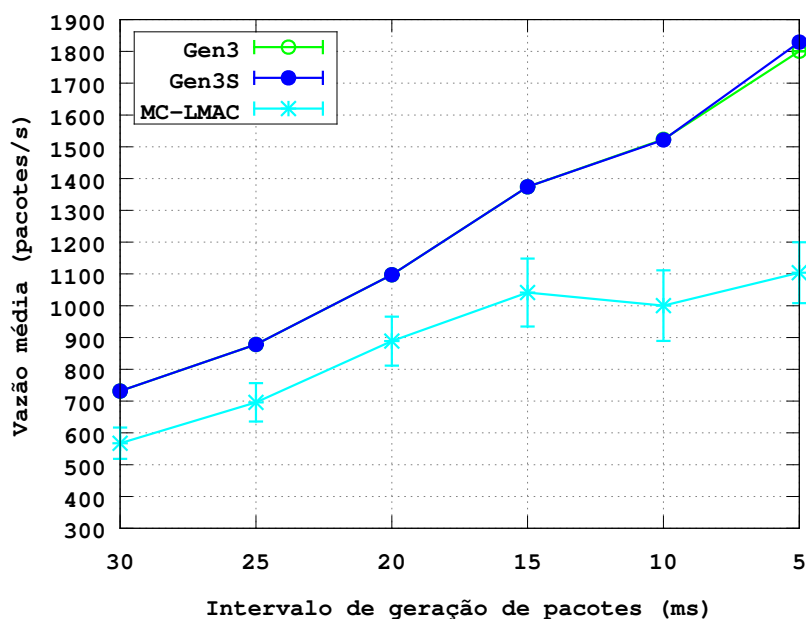
Já na topologia L23, cujo gráfico é apresentado na Figura 5.24, observa-se um crescimento significativo da vazão com o MC-LMAC, que ainda assim apresentou valores inferiores ao Gen3. Como discutido na seção anterior, o uso de MXC trouxe uma vantagem irrisória ao protocolo em topologias lineares, dado que apenas o *sink* possuía mais de um nó filho.

Figura 5.23 – Vazão média de pacotes na topologia L12 com tráfego CBR.



Fonte: do autor (2017).

Figura 5.24 – Vazão média de pacotes na topologia L23 com tráfego CBR.



Fonte: do autor (2017).

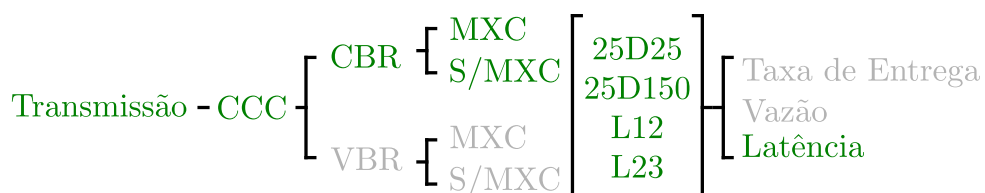
### 5.2.3 Latência - CBR

Os resultados avaliados nessa seção são destacados na Figura 5.25. Para analisar a latência foi definido um padrão de tráfego específico, com intervalo de 100 ms entre pacotes. Esse intervalo foi utilizado para garantir taxa de entrega de 100% em todos os casos, dado que



a perda de pacotes poderia influenciar na análise. Como todos os pacotes convergem para o *sink*, aqueles vindos de nós mais distantes e que fossem descartados não seriam contabilizados. Dessa forma, devido ao descarte seriam utilizados na análise mais pacotes provenientes de nós próximos ao *sink* do que de nós distantes, dando uma noção errônea dos tempos de latência.

Figura 5.25 – Cenário de transmissão avaliado: latência com tráfego CBR.



Fonte: do autor (2017).

O simulador armazena o tempo de criação de cada pacote. A cada pacote recebido, o *sink* armazena o tempo decorrido desde sua criação até sua recepção, subtraindo o tempo de criação do pacote do tempo atual de simulação.

Os dados de latência foram avaliados em histogramas com distribuições de frequências. Cada classe contém a quantidade média de pacotes recebidas dentro de intervalos de latência definidos no eixo x. O primeiro passo da análise foi contabilizar as frequências de cada classe em cada repetição, e posteriormente analisar a média de pacotes em cada uma.

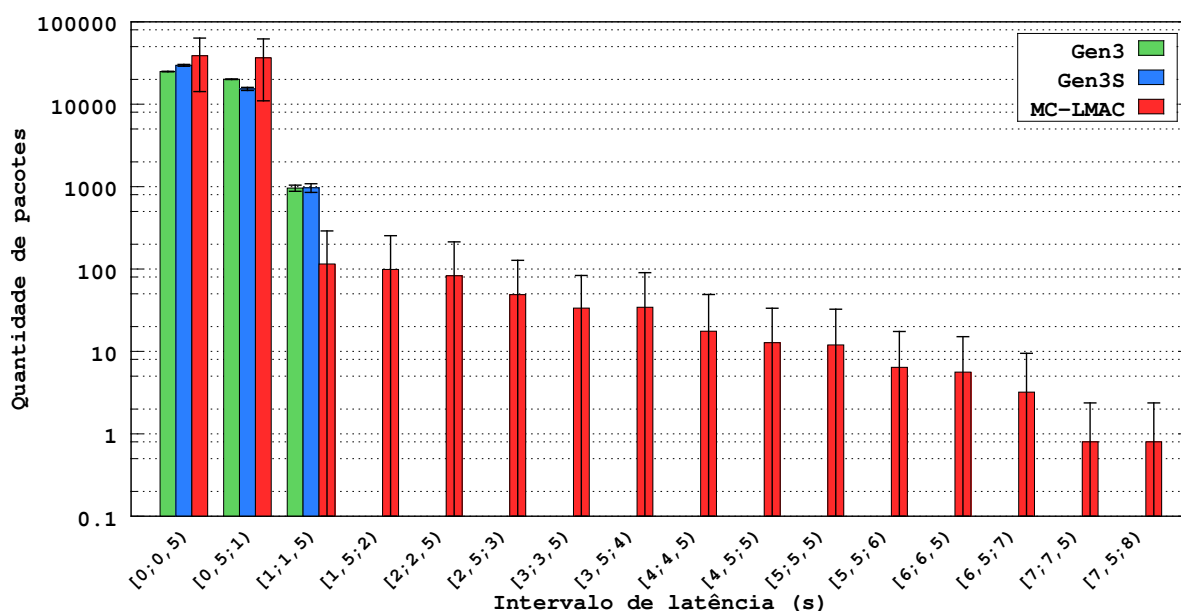
Os gráficos de latência estão apresentados em escala logarítmica dado que a diferença de valores, geralmente entre as classes de menor e maior latência, poderia dificultar a visualização dos resultados.

O gráfico da Figura 5.26 apresenta os resultados obtidos pelos protocolos na topologia 25D25. O Gen3 foi capaz de entregar todos os pacotes com latências entre 0 e 1,5 segundos, com maior frequência na classe de 0 a 0,5 segundos. Por outro lado, a entrega de pacotes no MC-LMAC ocorreu majoritariamente nas classes entre 0 e 1 segundos, mas também entregou pacotes com latências na classe de 7,5 a 8 segundos.

Na topologia 25D150, que tem os resultados apresentados no gráfico da Figura 5.27, ambos os protocolos entregaram todos os pacotes em menos de 1,1 segundos, com vantagem para o MC-LMAC, que entregou a maior parte dos pacotes entre 0 e 0,2 segundos. Como previsto, o Gen3 apresentou menor latência com o uso de MXC.

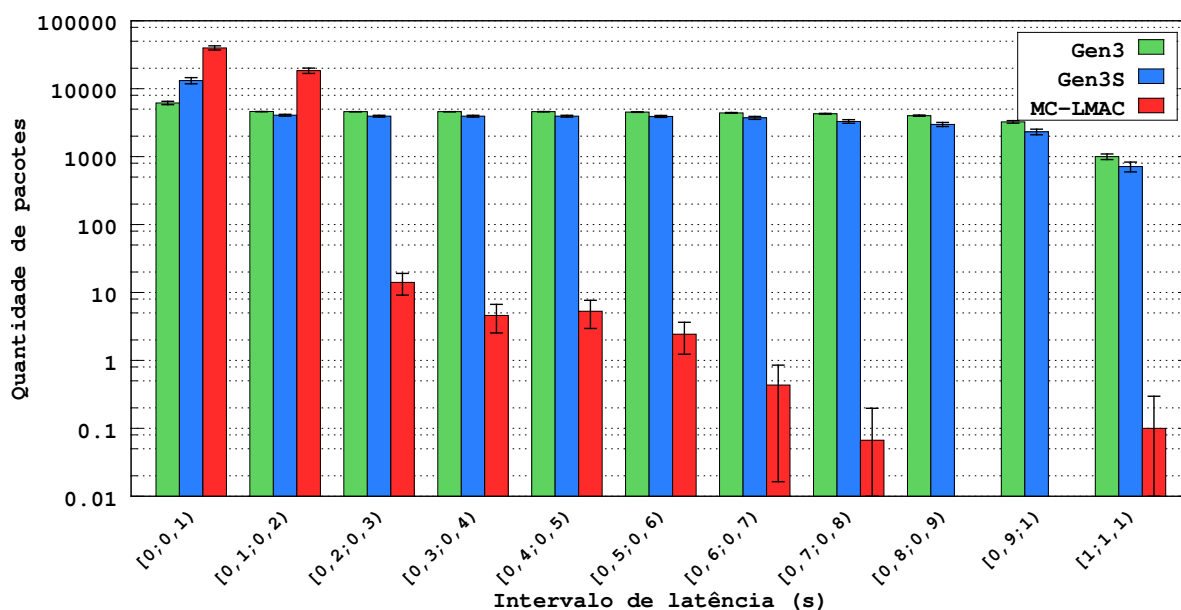
Por necessitar de uma quantidade menor de *slots*, resultando em um *frame* de menor duração, o MC-LMAC tem vantagem significativa em relação ao Gen3 nesse caso de teste.

Figura 5.26 – Distribuição de latências na topologia 25D25 com tráfego CBR.



Fonte: do autor (2017).

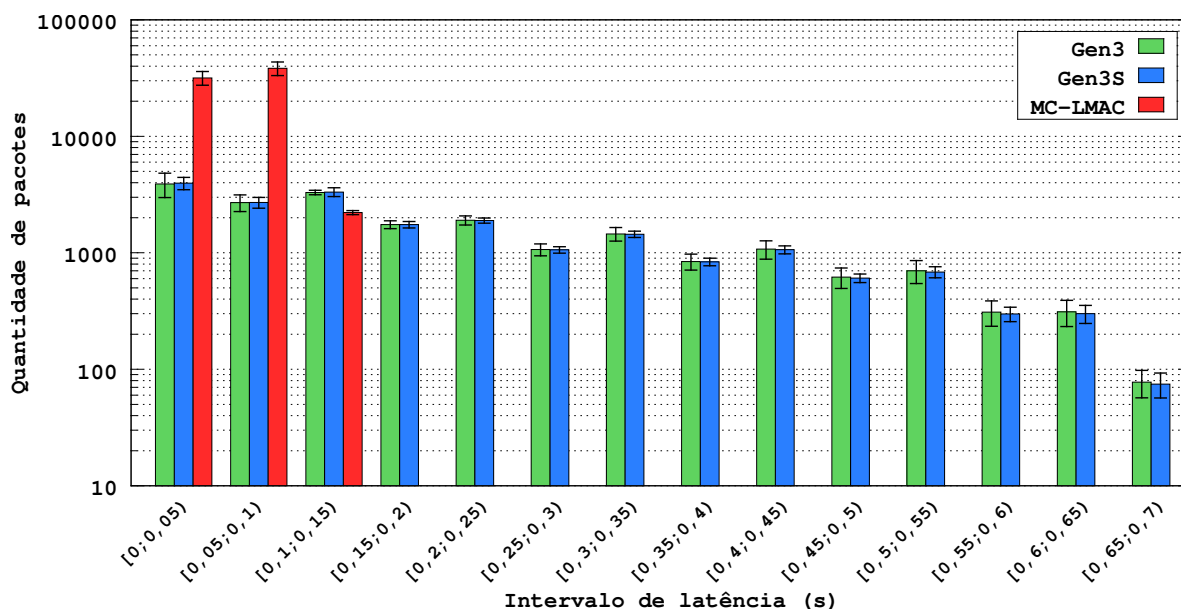
Figura 5.27 – Distribuição de latências na topologia 25D150 com tráfego CBR.



Fonte: do autor (2017).

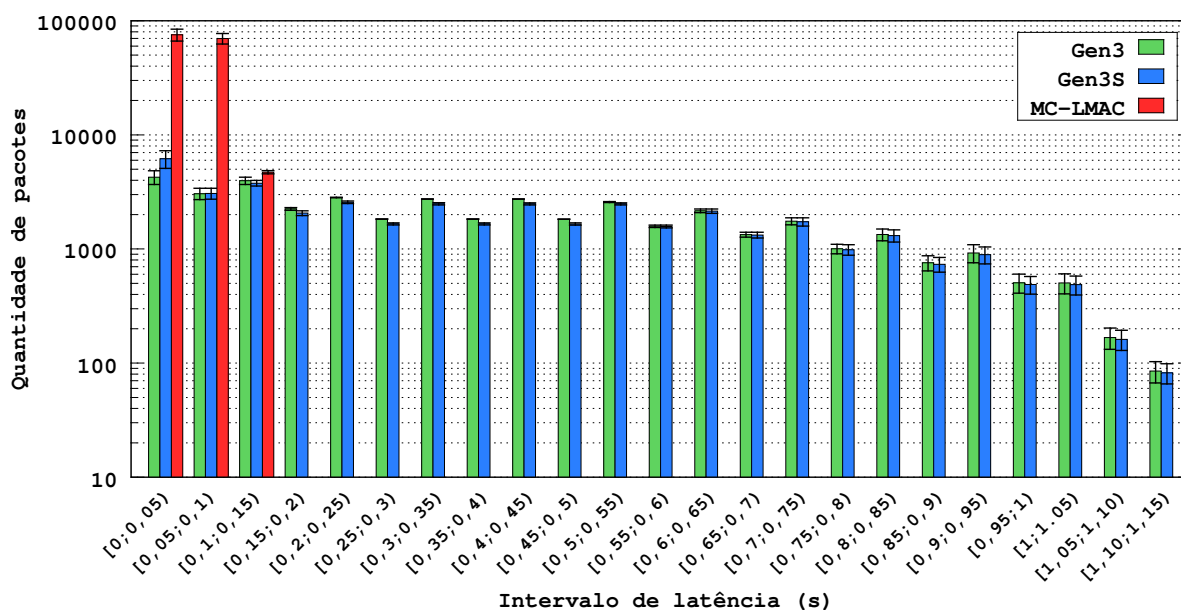
Nos testes em topologias lineares, a quantidade menor de *slots* do MC-LMAC novamente resultou em latências consideravelmente inferiores às obtidas pelo Gen3 com e sem MXC. Nos gráficos da Figuras 5.28 e 5.29 são apresentados os resultados nas topologias L12 e L23 respectivamente.

Figura 5.28 – Distribuição de latências na topologia L12 com tráfego CBR.



Fonte: do autor (2017).

Figura 5.29 – Distribuição de latências na topologia L23 com tráfego CBR.



Fonte: do autor (2017).

Enquanto o MC-LMAC entregou todos os pacotes com latências inferiores a 0,15 segundos, o Gen3 entregou pacotes com latências na classe de 0,65 a 0,7 segundos na topologia L12, e na classe de 1,10 a 1,15 segundos na topologia L23.

Ainda que o MC-LMAC seja capaz de entregar pacotes com latência significativamente inferior ao Gen3, em condições de tráfego mais intenso, como analisadas nas seções anteriores,

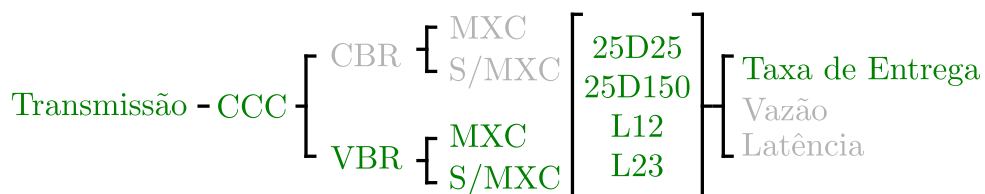
o MC-LMAC apresenta maior perda e menor vazão. Para uma aplicação em execução sobre o MC-LMAC, a alta taxa de perdas e a menor vazão de dados pode anular os efeitos positivos da latência, visto que mais retransmissões de pacotes podem ser necessárias.

Outro fator a ser considerado foi a necessidade de executar testes preliminares para determinar a configuração que permitira ao MC-LMAC obter os melhores resultados. A alteração da densidade dessas redes, com as configurações utilizadas nesses testes, poderia inviabilizar a associação de todos os nós ou implicar na deterioração da performance. Por outro lado, o Gen3 adapta a quantidade de *slots* de acordo com a quantidade de nós na rede, sendo capaz de oferecer conectividade a todos os nós independente da densidade de nós.

#### 5.2.4 Taxa de entrega - VBR

Os resultados avaliados nessa seção são destacados na Figura 5.30. No padrão de tráfego VBR os nós de destino dos pacotes eram definidos a cada nova rajada de mensagens.

Figura 5.30 – Cenário de transmissão avaliado: taxa de entrega com tráfego VBR.



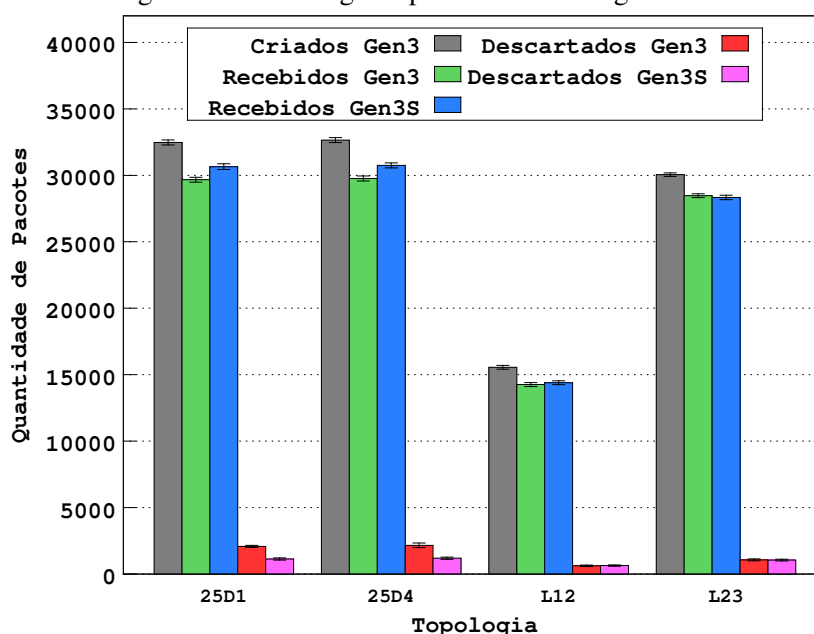
Fonte: do autor (2017).

O protocolo WiseRoute, utilizado para o roteamento de pacotes no MC-LMAC, opera apenas em modo de convergência, no qual todos os nós transmitem para um mesmo destino. Por conta disso, nos testes com tráfego VBR foi utilizado somente o protocolo Gen3, sendo feita sua avaliação com MXC ativada e desativada.

O gráfico da Figura 5.31 traz os resultados de entrega de pacotes para todas as topologias testadas. Da mesma forma que no tráfego CBR, em topologias lineares não foram observadas variações significativas com o uso de MXC. Nas topologias 25D25 e 25D150 foi observado um ligeiro incremento da taxa de entrega e, conseqüentemente, redução do descarte de pacotes com a MXC ativada.

Os descartes de pacotes ocorreram majoritariamente em nós com poucos *frames*, que necessitavam aguardar mais tempo para serem escalonados novamente. Ainda assim, esse descarte foi inferior a 10% dos pacotes. O aumento da capacidade de *buffer* dos nós permitiria que

Figura 5.31 – Entrega de pacotes com tráfego VBR.



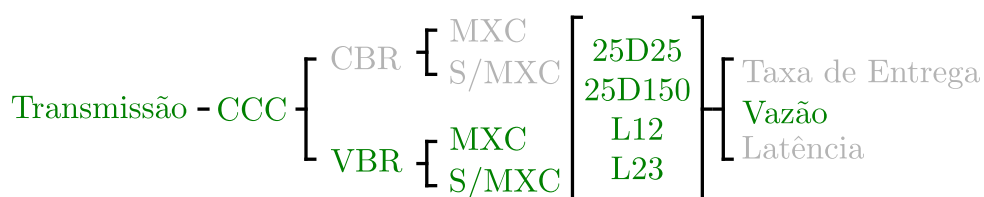
Fonte: do autor (2017).

eles armazenassem mais pacotes enquanto aguardam seus *slots* novamente, reduzindo a taxa de descarte.

### 5.2.5 Vazão da rede - VBR

Os resultados avaliados nessa seção são destacados na Figura 5.32. Neste cenário foi avaliada a vazão média de pacotes em cada nó. Como o tráfego é disperso, a tendência é que o valor de vazão seja menor do que o observado nos casos de tráfego constante.

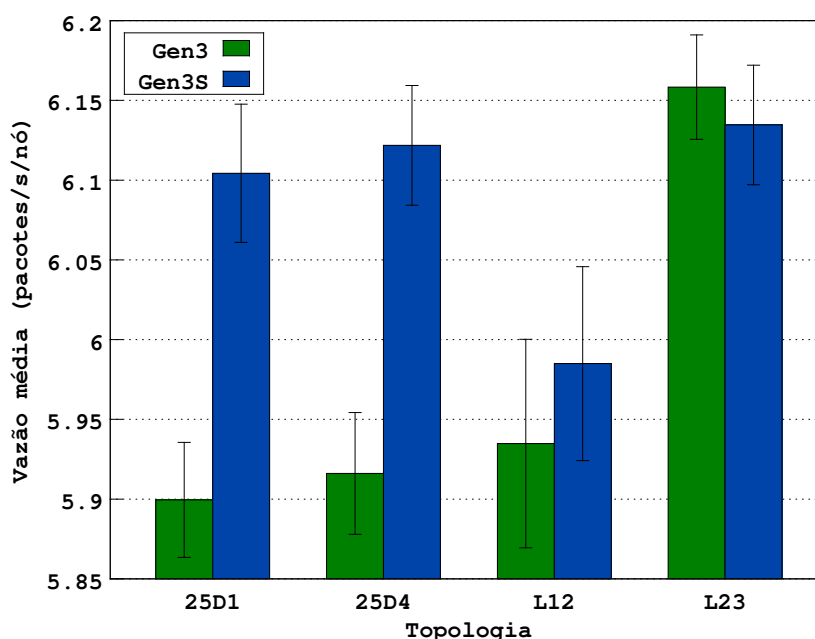
Figura 5.32 – Cenário de transmissão avaliado: vazão com tráfego VBR.



Fonte: do autor (2017).

O gráfico da Figura 5.33 apresenta os resultados de vazão obtidos nas topologias testadas. Como esperado, em todos os casos a vazão média por nó foi inferior a 6,2 pacotes por segundo. Nas topologias 25D25 e 25D150 é possível verificar um sutil incremento da vazão com o uso de MXC, enquanto nas topologias lineares, L12 e L23, a multiplexação não resultou em ganho de desempenho real.

Figura 5.33 – Vazão média de pacotes com tráfego VBR.

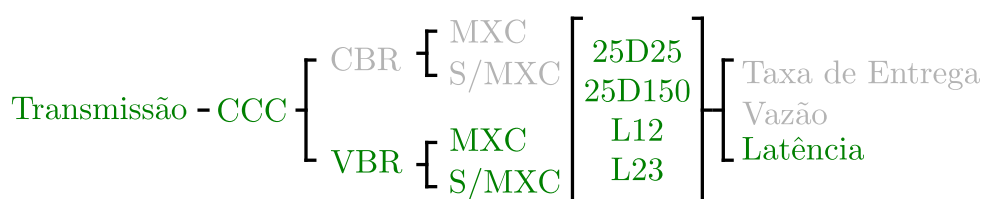


Fonte: do autor (2017).

### 5.2.6 Latência - VBR

Os resultados avaliados nessa seção são destacados na Figura 5.34. No caso de tráfego convergente, como utilizado no padrão CBR, a análise da latência pode ser prejudicada pela perda de pacotes, favorecendo aqueles que são transmitidos por nós próximos ao ponto de convergência das mensagens.

Figura 5.34 – Cenário de transmissão avaliado: latência com tráfego VBR.



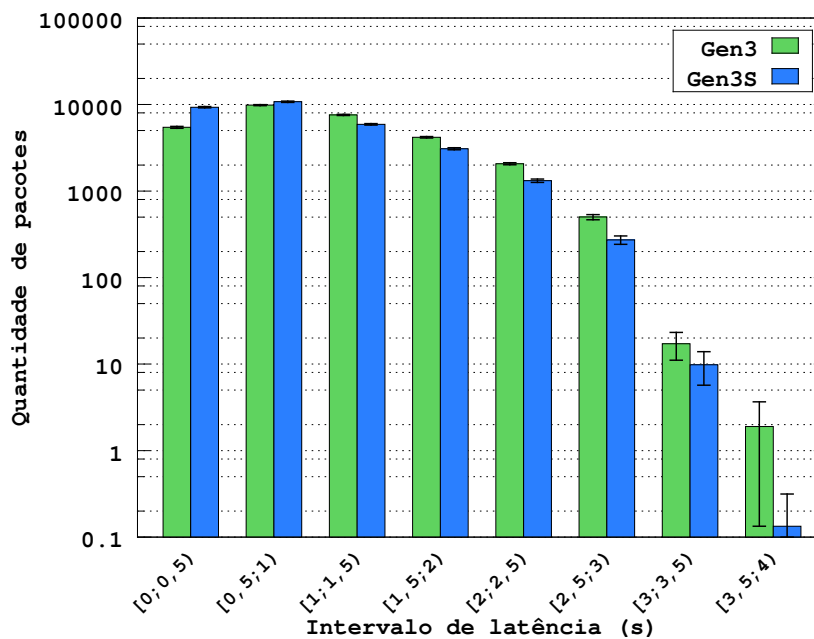
Fonte: do autor (2017).

Todavia, como no tráfego VBR os pacotes são endereçados a nós aleatórios da rede, nenhum nó está sempre próximo do destino de seus pacotes. Além disso, essa aleatoriedade dos destinatários reduz o afunilamento do tráfego de mensagens. Considerando essas condições, é possível realizar uma análise consistente da latência com tráfego VBR, mesmo com a taxa de entrega inferior a 100%.

Na Figura 5.35, o gráfico mostra os resultados obtidos na topologia 25D25. O uso de MXC permitiu a redução da latência, com o Gen3S entregando mais pacotes dentro das classes

de menor latência. Todos os pacotes foram entregues através da rede com tempos inferiores a 4 segundos, com incidência consideravelmente maior nas classes de menor latência.

Figura 5.35 – Distribuição de latências na topologia 25D25 com tráfego VBR.



Fonte: do autor (2017).

O comportamento do protocolo é semelhante para a topologia 25D150, como mostrado no gráfico da Figura 5.36, com maior incidência de tráfego na classe de latência entre 500 ms e 1 s.

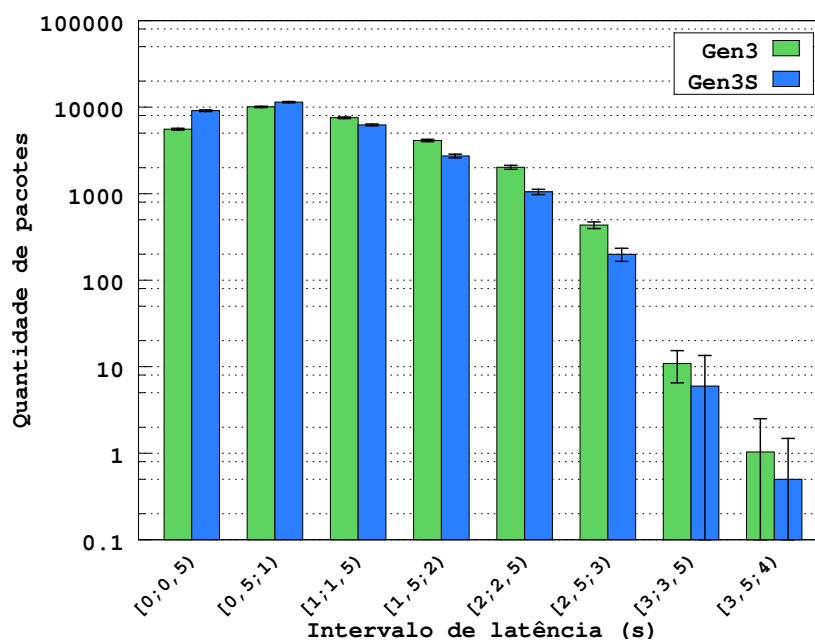
Na topologia linear L12 não foram observados ganhos de performance com o uso de MXC, assim como visto no tráfego CBR. Os resultados apresentados no gráfico da Figura 5.37 mostram maior frequência de pacotes nas classes de menor latência, principalmente na classe entre 0 s e 100 ms. Todos os pacotes foram entregues com tempos inferiores a 1,4 s.

Com a topologia linear L23, observa-se novamente que não houve melhoria na performance com o uso de MXC. Como observado no gráfico da Figura 5.38, as classes de latência utilizadas foram as mesmas das topologias de 25 nós, com maior frequência de entrega nas classes de menor valor.

### 5.2.7 Considerações finais

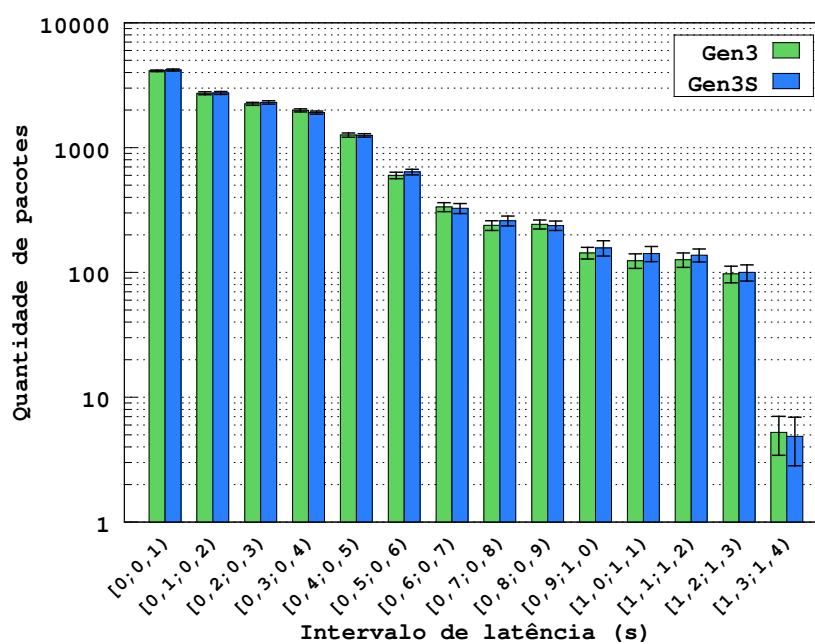
Os resultados obtidos nos testes de transmissão mostraram larga vantagem do Gen3 sobre o MC-LMAC em taxa de entrega e vazão, tanto com MXC ativada quanto desativada. Mesmo nas redes esparsas, em que era possível utilizar quantidades menores de *slots* no MC-

Figura 5.36 – Distribuição de latências na topologia 25D150 com tráfego VBR.



Fonte: do autor (2017).

Figura 5.37 – Distribuição de latências na topologia L12 com tráfego VBR.



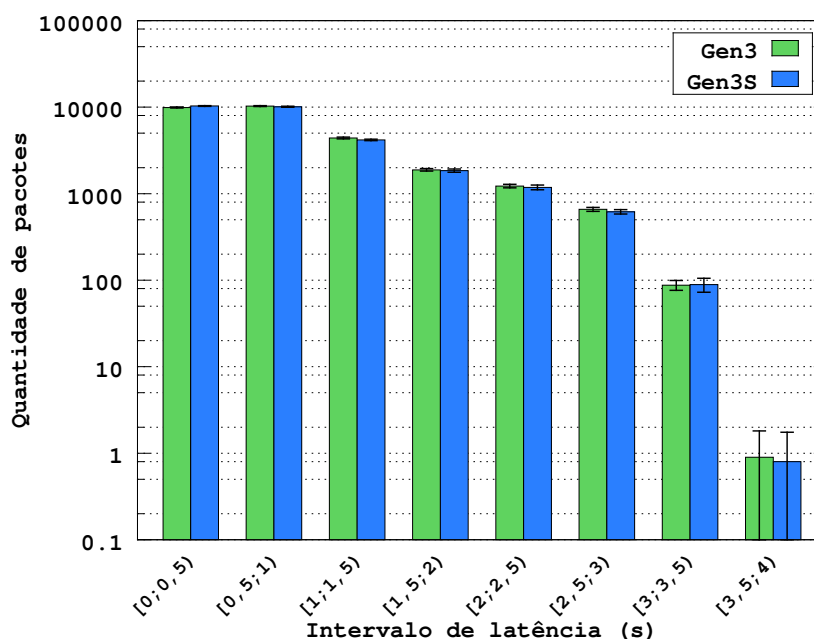
Fonte: do autor (2017).

LMAC, os resultados do Gen3 foram melhores nessas métricas em todos os intervalos de tráfego.

Nas topologias lineares, o uso de MXC não resultou em ganho de performance no Gen3, em relação à configuração sem MXC. Isso ocorre devido à forma como a MXC foi implemen-



Figura 5.38 – Distribuição de latências na topologia L23 com tráfego VBR.



Fonte: do autor (2017).

tada, tirando proveito apenas quando um nó possui dois ou mais nós filhos. Nas topologias 25D25 e 25D150, a MXC se mostrou eficaz em aumentar a vazão de pacotes e a taxa de entrega, principalmente com o crescimento da intensidade do tráfego. Além disso, o uso de MXC permitiu a entrega de mais pacotes em classes de latência mais baixas.

Na comparação da latência, o MC-LMAC foi capaz de entregar quantidades de pacotes significativamente superiores ao Gen3, principalmente nas topologias mais esparsas que permitiam o uso de uma menor quantidade de *slots*. No entanto, como mencionado anteriormente, devido à alta taxa de descarte essa menor latência pode ser anulada pela necessidade de retransmissão e pela baixa vazão oferecidas pelo MC-LMAC.

Com tráfego VBR foi possível verificar a capacidade de operação do Gen3 com tráfego disperso, e não apenas com tráfego convergente. O Gen3 apresentou taxas de descarte inferiores a 10% dos pacotes, e as perdas de pacotes foram inferiores a 0,0004%. A latência observada com tráfego VBR foi superior em relação ao tráfego CBR, mas sempre inferior a 4 segundos nas topologias 25D25, 25D150 e L23, e inferior a 1,4 segundos na topologia L12.

### 5.3 Mobilidade

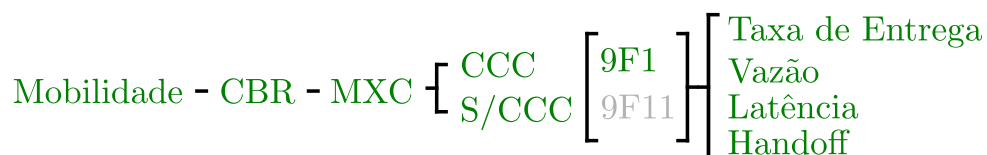
Diferentemente dos demais testes, a avaliação dos resultados dos experimentos de mobilidade será dividida pelas topologias utilizadas, com a análise dos resultados obtidos com a topologia 9F1 e depois com a topologia 9F11. Para cada uma foram avaliadas as métricas de vazão, taxa de entrega, latência e *handoff*.

Nas análises de mobilidade não foi incluído o MC-LMAC, sendo avaliado apenas o Gen3 com variações de configuração. Como a implementação do MC-LMAC neste trabalho depende do WiseRoute para o roteamento de pacotes, a análise do seu desempenho seria afetada pelo protocolo de roteamento. O WiseRoute precisa transmitir pacotes em *broadcast* para a descoberta de rotas em intervalos predefinidos, essas rotas são descartadas e novos *broadcasts* são realizados para suportar as variações de topologia.

#### 5.3.1 Topologia 9F1

A Figura 5.39 mostra o cenário avaliado nessa seção. A Tabela 5.4 traz os parâmetros adicionais utilizados nos testes com a topologia 9F1. Por ser uma rede esparsa e visando acelerar a associação dos nós, o parâmetro *Settle Cycles* foi ajustado para 1, enquanto a quantidade de canais foi mantida em 6.

Figura 5.39 – Cenário de mobilidade avaliado: topologia 9F1.



Fonte: do autor (2017).

Tabela 5.4 – Parâmetros adicionais do teste de mobilidade na topologia 9F1.

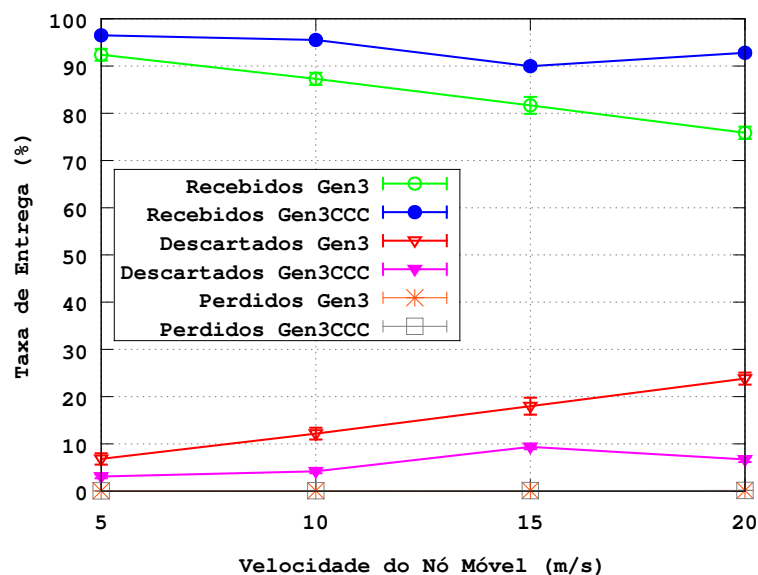
Parâmetro	Valor
<i>Settle Cycles</i>	1
Canais	6
Intervalo entre envio de pacotes	20 ms
Velocidade do nó móvel (m/s)	5, 10, 15, 20

Fonte: do autor (2017).

O gerador de tráfego CBR transmitiu pacotes em intervalos de 20 ms. O nó móvel foi configurado para ter velocidade constante em cada simulação e foram executadas simulações com velocidades de 5, 10, 15 e 20 m/s. Essas velocidades são equivalentes a 18, 36, 54 e 72 km/h respectivamente.

O primeiro resultado a ser analisado é a taxa de entrega, cujos resultados são apresentados no gráfico da Figura 5.40. Utilizando CCC a taxa de entrega foi maior em todos os casos, pois essa configuração facilita a reassociação do nó à rede quando ele se desconecta de um ponto.

Figura 5.40 – Taxa de entrega na topologia 9F1.



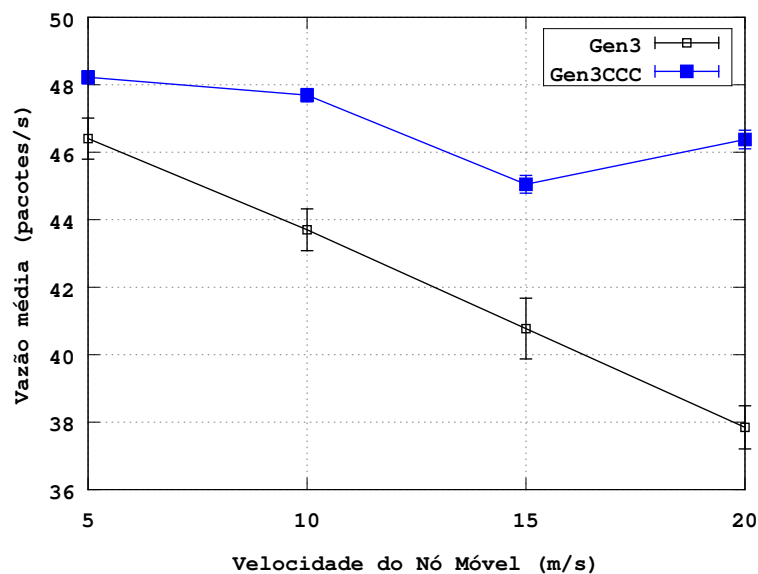
Fonte: do autor (2017).

Apesar da taxa de descarte ter sido próxima a 10% no pior caso com CCC, e de cerca de 25% sem CCC, a taxa de perdas foi inferior a 0,3% em todos os casos. Essa informação é relevante para a análise, pois o descarte pode ser atenuado pelo aumento da capacidade de *buffer* dos nós, de modo que possam armazenar mais pacotes durante os períodos de desconexão.

Outra forma de atenuar o descarte de pacotes, como sugerido na RFC3819 (REINER; FAIRHURST; WOOD, 2004), é que haja uma interface de comunicação entre a camada MAC e a camada superior para que a camada MAC possa informar que seu *buffer* está cheio, e permita que os serviços das camadas superiores interrompam o envio de novos pacotes.

Os resultados de vazão são mostrados no gráfico da Figura 5.41 e novamente mostram vantagem do uso de CCC na operação do protocolo. A quantidade de pacotes por segundo entregue ao *sink* foi maior com todas as velocidades do nó móvel utilizando a configuração com CCC. A facilidade de reassociação permitida pelo CCC resultou em melhor performance do protocolo.

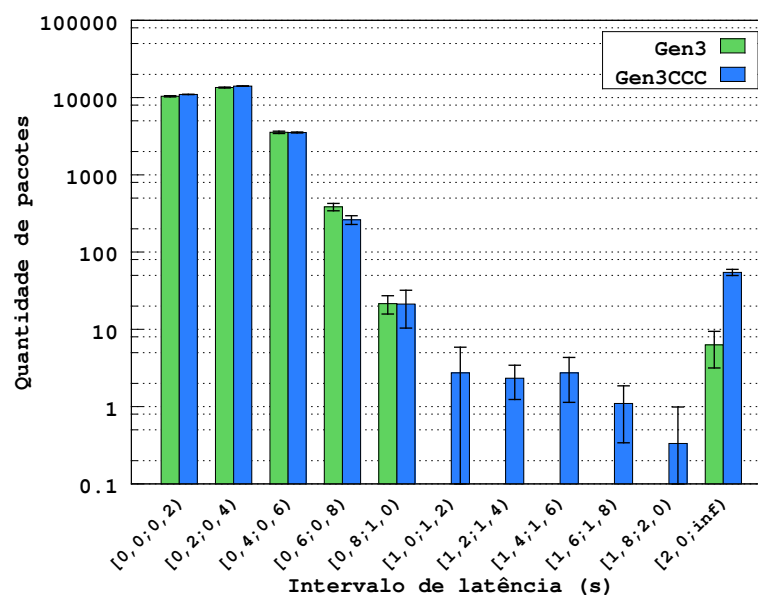
Os valores de latência obtidos são apresentados no gráfico da Figura 5.42. A classe de latências entre 200 ms e 400 ms apresentou a maior frequência de pacotes. A vasta maioria

Figura 5.41 – Vazão média no *sink* com a topologia 9F1.

Fonte: do autor (2017).

dos pacotes foi entregue em tempo inferior a 600 ms, enquanto em alguns casos esporádicos ocorreu a entrega de pacotes com tempos superiores a 1 segundo.

Figura 5.42 – Distribuição de latências na topologia 9F1.



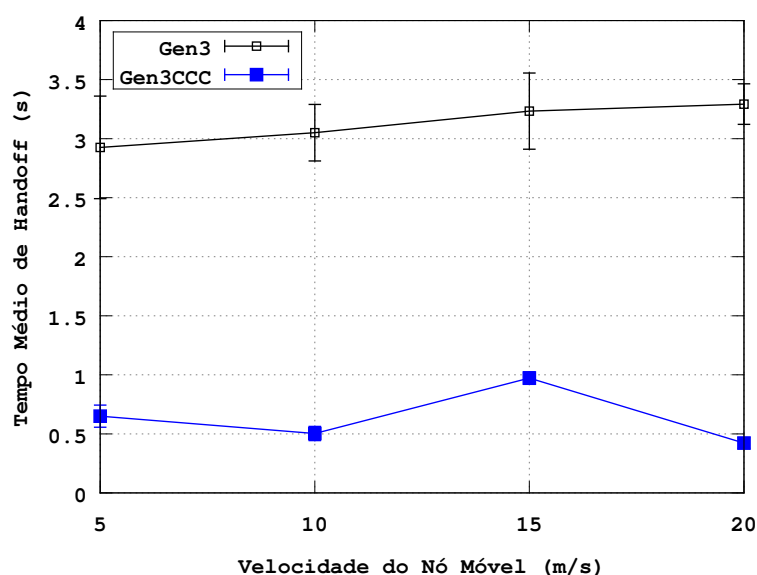
Fonte: do autor (2017).

No gráfico de latência é possível verificar a influência da taxa de entrega na análise da latência. Como na configuração sem CCC o descarte de pacotes foi maior, os pacotes mais antigos do *buffer* eram descartados, eliminando aqueles que apresentariam maior latência. Por

permitir maior vazão na configuração com CCC, os pacotes mais antigos do *buffer* têm menos probabilidade de serem descartados do que sem o uso de CCC. Assim, pacotes mais antigos chegam ao *sink* e elevam a média de latência observada. Pela análise do gráfico é possível verificar que na configuração com CCC houve maior incidência de pacotes nas classes de maior latência do que na configuração sem CCC.

O tempo médio de *handoff*, mostrado no gráfico da Figura 5.43, tem relação direta com os resultados verificados de vazão e taxa de entrega. Quanto menor o tempo médio de *handoff*, maior a vazão de pacotes e maior a taxa de entrega. Enquanto desconectado da rede e tentando uma nova reassociação, o nó é incapaz de dar vazão aos pacotes recebidos e depende de sua capacidade de *buffer* para evitar a perda de pacotes.

Figura 5.43 – Tempo médio de *handoff* na topologia 9F1.



Fonte: do autor (2017).

Sem o uso de CCC, o nó deve varrer todos os canais utilizados em busca de pacotes de associação, o que atrasa sua reentrada. Isso não acontece com o uso de CCC, já que os pacotes de associação são transmitidos sempre no canal comum, restando ao nó desconectado ouvir neste único canal para receber pacotes de associação.

### 5.3.2 Topologia 9F11

A Figura 5.44 apresenta o cenário avaliado nessa seção. A Tabela 5.5 traz os parâmetros adicionais para a topologia 9F11.

Figura 5.44 – Cenário de mobilidade avaliado: topologia 9F11.

$$\text{Mobilidade - CBR - MXC} \left[ \begin{array}{c} \text{CCC} \\ \text{S/CCC} \end{array} \right] \left[ \begin{array}{c} 9F1 \\ 9F11 \end{array} \right] \left[ \begin{array}{c} \text{Taxa de Entrega} \\ \text{Vazão} \\ \text{Latência} \\ \text{Handoff} \end{array} \right]$$

Fonte: do autor (2017).

Tabela 5.5 – Parâmetros adicionais do teste de mobilidade na topologia 9F11.

Parâmetro	Valor
<i>Settle Cycles</i>	4
Canais	6
Intervalo entre envio de pacotes	20 ms
Velocidade dos nós móveis	Valor aleatório entre 1 e 5 m/s
Variação de direção dos nós móveis	Valor aleatório entre 0° e 45°

Fonte: do autor (2017).

Para esse teste foram utilizados 20 nós, sendo 9 dispostos de forma regular cobrindo uma área e 11 nós dispostos aleatoriamente com mobilidade. Em intervalos entre 500 ms e 2 s os nós decidiam uma velocidade e uma direção para sua movimentação, sendo a nova velocidade um valor entre 1 e 5 m/s, e a direção um valor entre 0° e 45°.

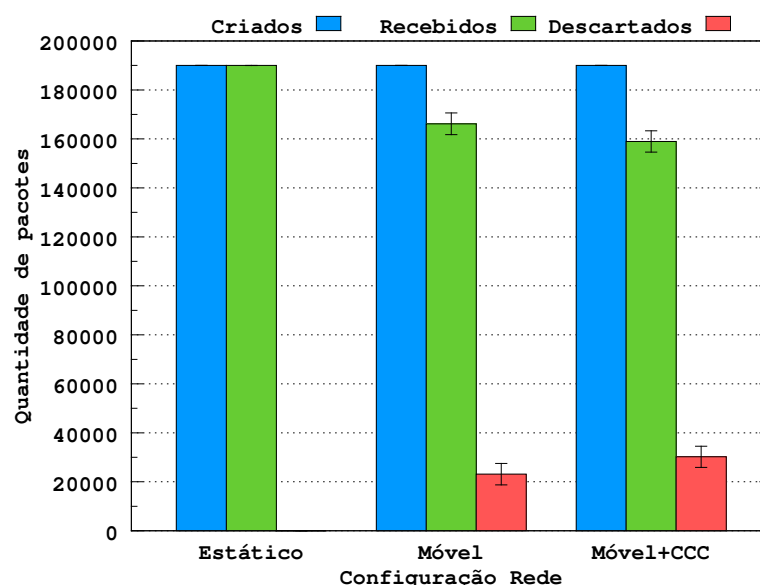
Como a disposição dos 11 nós móveis é aleatória, o parâmetro *Settle Cycles* foi ajustado para 4, seguindo as observações feitas na análise de configuração da rede.

O principal objetivo desse teste é verificar a influência da mobilidade na operação do protocolo. Para isso, primeiramente foram realizados testes sem a movimentação dos nós. Esse caso serve como parâmetro para a comparação da operação do protocolo com mobilidade, além de permitir verificar a operação do protocolo em redes dispostas irregularmente. Posteriormente, foram executados os testes com os 9 nós fixos e 11 móveis, os dados dos testes com e sem mobilidade foram analisados conjuntamente para determinar os efeitos da mobilidade no desempenho do protocolo.

O gráfico da Figura 5.45 traz os resultados de taxa de entrega obtidos na topologia 9F11. Na configuração estática não foi observada diferença na taxa de entrega nas configurações com e sem CCC, com entrega superior a 99,99% dos pacotes. Por não ter tido variação, foram utilizados como referência apenas os dados com a topologia estática sem CCC.

Com essa topologia, a configuração móvel com CCC apresentou taxa de entrega ligeiramente inferior à da configuração sem CCC. A taxa de perda de pacotes não foi incluída no

Figura 5.45 – Taxa de entrega na topologia móvel 9F11.



Fonte: do autor (2017).

gráfico por ser inferior a 0,05% no pior caso, com média de 86,36 ( $\pm 5,99$ ) pacotes perdidos na configuração móvel com CCC.

Os resultados de vazão e *handoff* são apresentados na Tabela 5.6. A vazão média, dada pela quantidade de pacotes entregues no *sink* por segundo, foi semelhante na rede estática com e sem CCC, atingindo cerca de 948 pacotes por segundo. Com mobilidade, a vazão caiu para cerca de 829 e 793 pacotes por segundo nas configurações sem CCC e com CCC respectivamente.

Tabela 5.6 – Resultados de vazão e *handoff* obtidos na topologia 9F11.

Métrica	Configuração	Valor
Vazão média no <i>sink</i> (pacotes por segundo)	Estática	948,19 ( $\pm 0,04$ )
	Estática+CCC	948,22 ( $\pm 0,04$ )
	Móvel	829,38 ( $\pm 22,06$ )
	Móvel+CCC	793,62 ( $\pm 21,81$ )
Tempo médio de <i>handoff</i> (segundos)	Estática	2,34 ( $\pm 0,75$ )
	Estática+CCC	2,97 ( $\pm 1,79$ )
	Móvel	12,91 ( $\pm 0,92$ )
	Móvel+CCC	8,42 ( $\pm 0,49$ )

Fonte: do autor (2017).

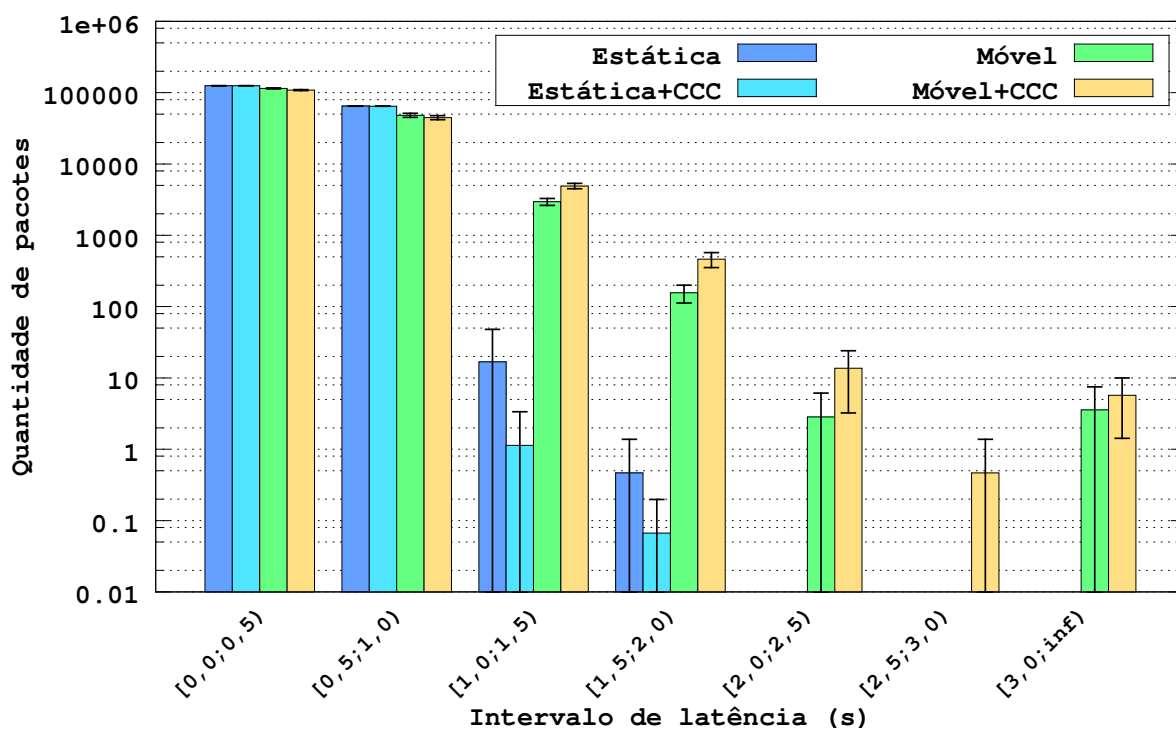
Contrariando a análise das demais métricas, o tempo médio de *handoff* observado na configuração sem CCC foi maior do que na configuração com CCC. Contudo, utilizando CCC a quantidade média de *handoffs* por nó foi de 6,22 ( $\pm 0,34$ ), enquanto sem CCC essa quantidade

foi de  $3,72 (\pm 0,21)$  *handoffs*. Isso indica que apesar de ter tempo médio de *handoff* menor, foram realizados mais *handoffs* na configuração com CCC.

A relação de maior quantidade e menor duração dos *handoffs* na configuração com CCC resultou em perda de performance, se comparado ao caso de menor quantidade e maior duração de *handoffs*, observado na configuração sem CCC.

Os resultados de latência para a topologia 9F11 são apresentados no gráfico da Figura 5.46. A grande maioria dos pacotes foi entregue com latências nas classes de 0 s a 500 ms, e 500 ms a 1 s. Enquanto nas configurações estáticas apenas uma pequena quantidade de pacotes foi entregue entre 1 e 2 segundos, a frequência de pacotes nessas classes foi significativamente superior nas configurações com mobilidade.

Figura 5.46 – Distribuição de latências na topologia 9F11.



Fonte: do autor (2017).

Contudo, a porcentagem de pacotes entregues com latências superiores a 1 segundo é inferior a 10%, e acima de 2 segundos essa porcentagem cai para menos de 1% dos pacotes.

### 5.3.3 Considerações finais

No primeiro teste de mobilidade, com a topologia 9F1, os resultados mostraram clara vantagem do uso de CCC, que permitiu maior taxa de entrega, maior vazão de pacotes no *sink*



e menores tempos médios de *handoff*. Quanto à latência, a configuração com CCC implicou em maior frequência de pacotes nas classes de latência mais baixa, mas também nas classes de latência mais alta. Isso pode ser explicado pela maior vazão obtida com esta configuração, que reduz o descarte e permite a entrega de pacotes que aguardavam em *buffer* há mais tempo.

Com a topologia 9F11 foram executados dois testes: o primeiro com a rede estática, e o segundo com a mobilidade habilitada. O primeiro teste serviu de parâmetro para avaliar o impacto negativo da mobilidade do segundo teste no desempenho do Gen3.

A mobilidade reduziu a taxa de entrega devido ao descarte de pacotes, enquanto as perdas foram inferiores a 0,05% no pior caso. Não foram observadas diferenças significativas entre a configuração com CCC e sem CCC para a taxa de entrega.

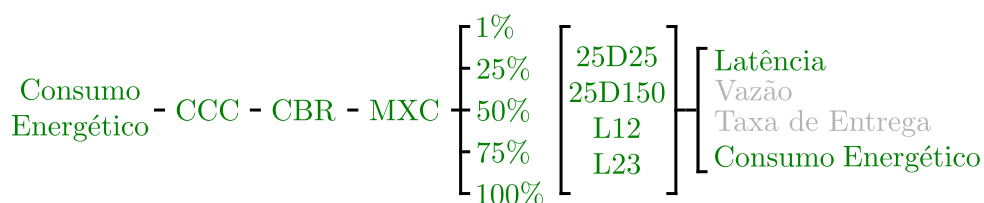
A mobilidade teve impacto negativo na vazão da rede, pois nos momentos de desconexão dos nós a quantidade de pacotes transmitidos pela rede é menor. Os tempos de *handoff* com o uso de CCC foram em média inferiores que na configuração sem CCC. No entanto, a quantidade de *handoffs* realizados com CCC foi maior, tendo influência negativa no desempenho do Gen3 com essa configuração.

Os resultados mostram que o Gen3 mantém taxas elevadas de entrega e vazão, mesmo com a mobilidade dos nós. Apesar da maior quantidade de pacotes entregues com latência superior a 2 segundos, a maioria dos pacotes foi entregue em tempos inferiores a 2 segundos.

## 5.4 Consumo de energia

Nessa seção são analisados os resultados dos testes de consumo de energia, destacados na Figura 5.47. Para esse teste foram definidas as análises de latência, vazão, taxa de entrega e consumo energético, utilizando tráfego CBR com intervalo de 50 ms entre pacotes. Com esse padrão de tráfego não foram observados resultados relevantes na taxa de entrega e na vazão com a variação dos parâmetros. Dessa forma, os resultados das métricas de vazão e taxa de entrega não foram incluídas para análise nessa seção.

Figura 5.47 – Cenário avaliado: consumo energético e MXC.



Fonte: do autor (2017).

No simulador o consumo de energia é medido de acordo com os estados do rádio. Os parâmetros de consumo para as simulações foram definidos de acordo com os valores encontrados na especificação do rádio nRF24101+, sendo descritos na Tabela 5.7. Ao final das simulações é possível saber quanta energia foi gasta por cada nó da rede. Os resultados apresentados nessa seção são referentes à média de energia consumida por todos os nós, verificada ao final do tempo de simulação.

São definidos 6 estados do rádio, que afetam seu consumo e sua operação. Mesmo configurado no modo desligado, o rádio ainda consome uma pequena carga, mas desprezível em relação ao consumo normal. Com o rádio ocioso, tanto em modo transmissor quanto em modo receptor, o consumo é o mesmo. Durante a recepção de pacotes ocorre o maior consumo de energia, com cerca de 40 mW, enquanto durante a transmissão de pacotes esse valor gira em torno de 37 mW. No modo de baixo consumo, *sleep*, são desligados apenas alguns componentes do rádio, permitindo que ele retome sua operação em menos tempo do que se estivesse totalmente desligado.

Nesses testes, além de comparar os protocolos, também foi avaliado o impacto de diferentes limiares de MXC no consumo energético. Um nó se mantém ouvindo o canal de transmissão somente quando a ocupação do *buffer* está acima do limiar definido, permitindo a economia de energia quando há baixo tráfego gerado pelo nó.

Tabela 5.7 – Parâmetros de consumo de energia do rádio nas simulações.

Parâmetro	Valor
Rádio desligado	0,003 mW
Receptor ocioso	1,06 mW
Receptor recebendo pacotes	40,6 mW
Rádio em modo <i>sleep</i>	0,07 mW
Transmissor ocioso	1,06 mW
Transmissor enviando pacotes	37,3 mW

Fonte: do autor (2017).

A Tabela 5.8 traz os parâmetros complementares utilizados, que são semelhantes aos utilizados nos testes de transmissão, exceto pelo limiar de MXC que tem as variações apresentadas na Figura 5.47. Os parâmetros do MC-LMAC são iguais aos utilizados no teste de transmissão (descritos na Tabela 5.3).

Tabela 5.8 – Parâmetros complementares do Gen3 para os testes de consumo energético.

Parâmetro	Valor
<i>Settle Cycles</i>	4
Quantidade de canais	6
Uso de CCC	Sim
Tipo de tráfego	CBR
Intervalo entre pacotes	50 ms

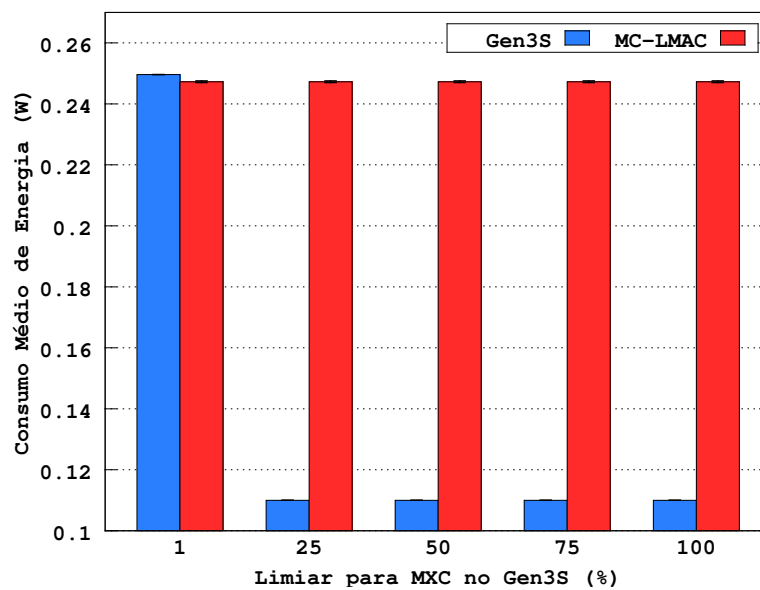
Fonte: do autor (2017).

Por não ter variação de parâmetros no MC-LMAC, os valores obtidos com ele são repetidos nas colunas dos gráficos de consumo para facilitar a comparação com as diferentes configurações de MXC do Gen3. O gráfico da Figura 5.48 mostra os resultados obtidos na topologia 25D25. Quando o limiar de MXC é de 1%, os nós entram em MXC quando possuem pelo menos 1 pacote em *buffer*, o que ocorre com maior frequência e faz com que o Gen3 tenha consumo médio ligeiramente maior que o MC-LMAC. A partir do limiar de 25% a MXC não foi utilizada, permitindo uma redução drástica no consumo de energia.

No gráfico da Figura 5.49, os resultados para a topologia 25D150 já mostram vantagem do Gen3 em todos os casos, mesmo com limiar de MXC igual a 1%. Na topologia L12, cujo gráfico é apresentado na Figura 5.50, a diferença de consumo entre os protocolos é menor, mas com estreita vantagem para o Gen3. Na topologia L23, os resultados apresentados no gráfico da Figura 5.51 mostram menor consumo do Gen3 em relação ao MC-LMAC em todos os casos.

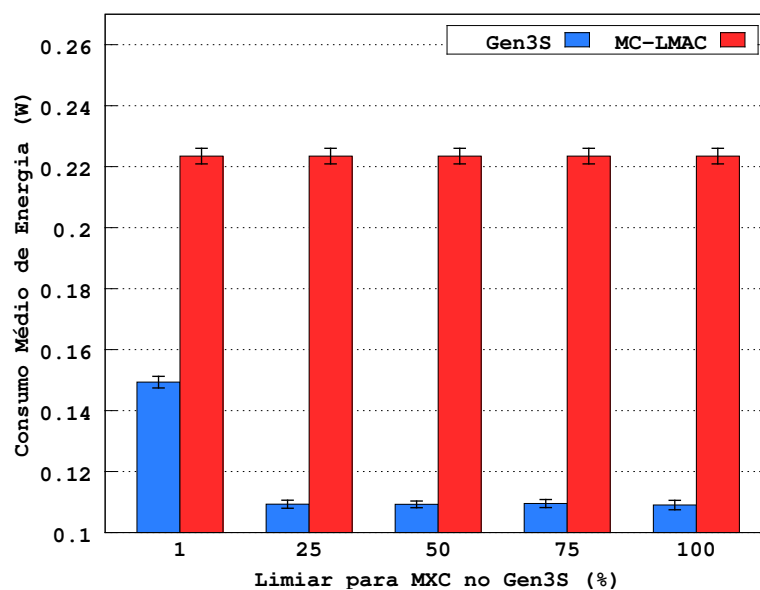
Diversos fatores influenciam no consumo energético dos protocolos e determinam a diferença nos resultados obtidos. Enquanto no MC-LMAC todos os nós devem ativar seus rádios em todos os *slots*, no Gen3 um nó pode manter seu rádio desligado se estiver fora de

Figura 5.48 – Consumo médio de energia na topologia 25D25.



Fonte: do autor (2017).

Figura 5.49 – Consumo médio de energia na topologia 25D150.

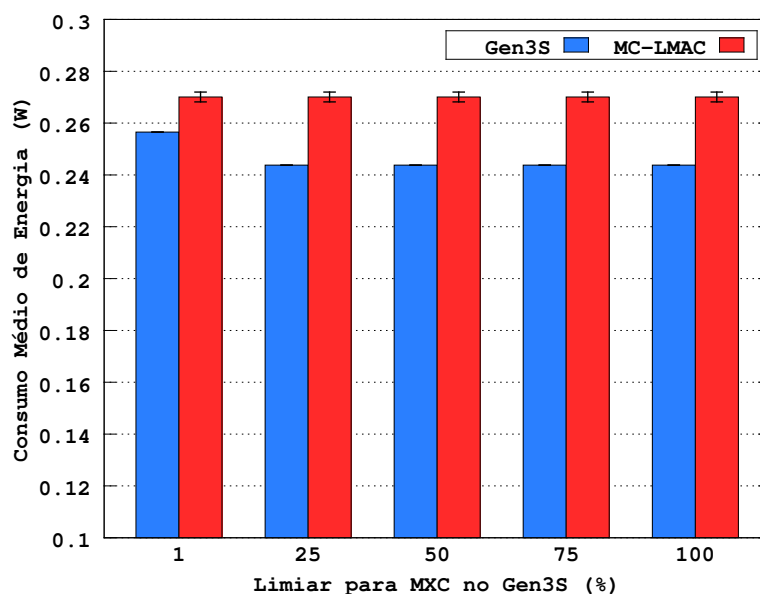


Fonte: do autor (2017).

seus *frames* e não estiver executando MXC. Essa diferença conceitual já permite a redução do consumo de energia no Gen3, principalmente pelos nós que possuem poucos *frames* e geram pouco tráfego.

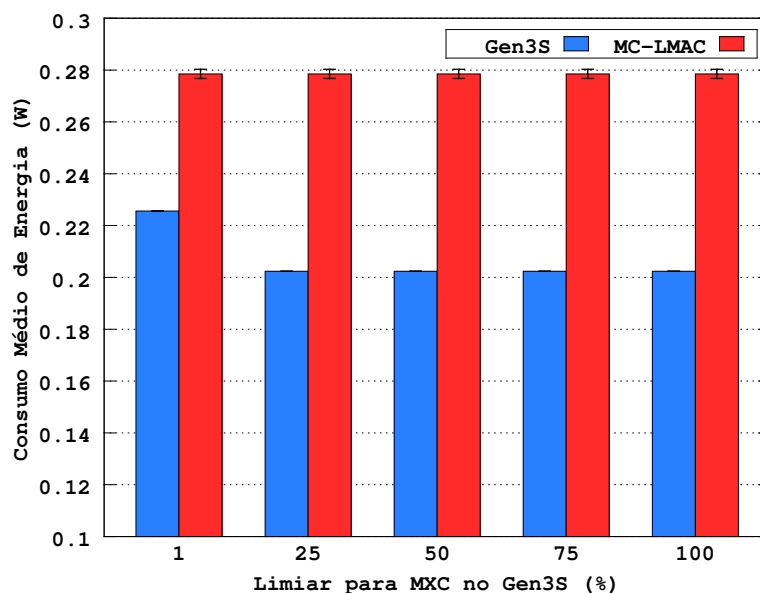
É importante ressaltar que os resultados apresentados são uma média do consumo de todos os nós da rede. No Gen3, os nós que possuem mais *frames*, como o *sink*, apresentam consumo consideravelmente maior que dos nós folha (que possuem apenas um *frame*). Nas

Figura 5.50 – Consumo médio de energia na topologia L12.



Fonte: do autor (2017).

Figura 5.51 – Consumo médio de energia na topologia L23.



Fonte: do autor (2017).

topologias regulares (25D25 e 25D150), a média de consumo é reduzida pela influência da maior quantidade de nós folha, enquanto nas topologias lineares (L12 e L23), o consumo dos nós mais próximos do *sink* eleva essa média.

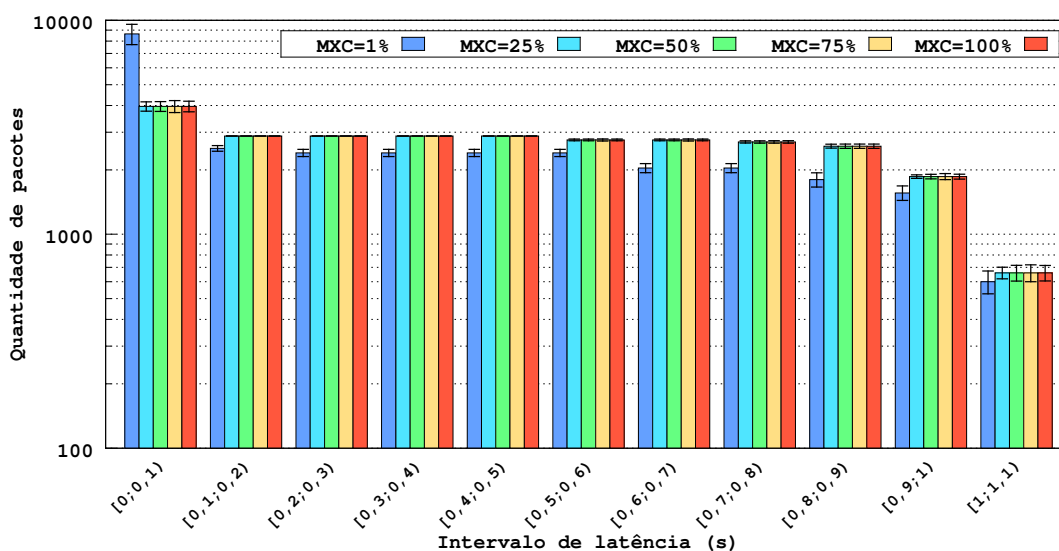
Como exemplo, na topologia L12, nó mais próximo do *sink* (nó 1) tem consumo ainda maior que o próprio *sink*. Enquanto este opera apenas no *slot 0* de todos os *frames*, o nó 1 opera

tanto no *slot* 0, quanto no *slot* 1. O nó 1 pode desligar seu rádio apenas no último *frame* do ciclo, e em um dos *slots* de seu último *frame*.

Ainda assim, durante os *slots* de operação, o nó 1 desativa seu rádio sempre que as comunicações dentro do *slot* são encerradas, tentando reduzir seu consumo ao mínimo necessário.

Em adição à análise de consumo energético, foi analisada também a latência obtida com o Gen3 com os diferentes valores de limiar para MXC. Foram observadas diferenças apenas com as topologias 25D25 e 25D150, mostradas nos gráficos das Figuras 5.52 e 5.53. Devido ao baixo tráfego, somente com limiar de 1% foram verificados ganhos de performance. Na topologia 25D25 a frequência de pacotes na classe de menor latência com o limiar de 1% foi superior a 8.000 pacotes, enquanto com os demais limiares esses valores foram de cerca de 3.900 pacotes.

Figura 5.52 – Latência com diferentes limiares de MXC na topologia 25D25.



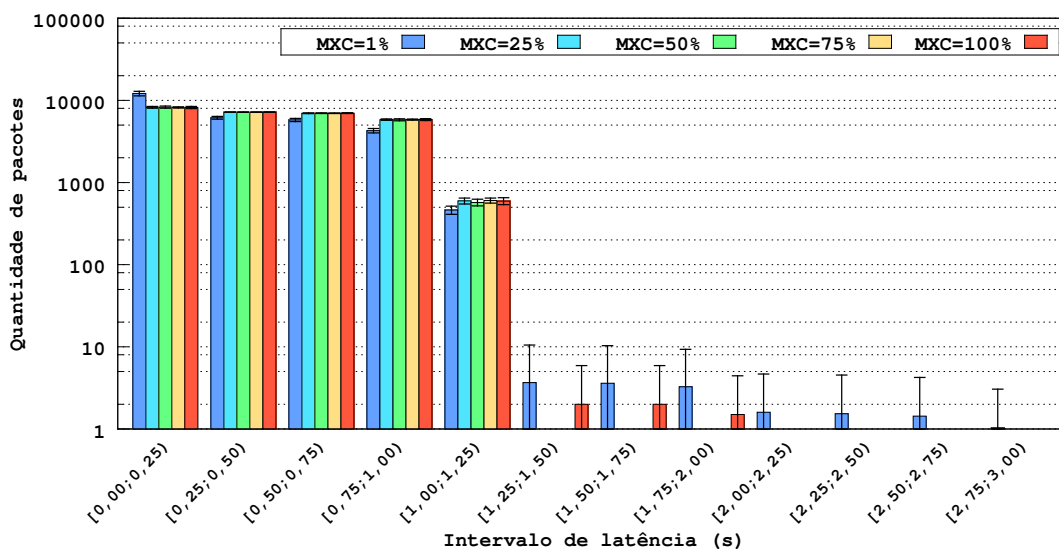
Fonte: do autor (2017).

Na topologia 25D150, essa diferença foi mais sutil, mas ainda com maior frequência de pacotes na classe de menor latência, com cerca de 12.000 pacotes com o limiar de 1% contra cerca de 8.200 pacotes nos demais limiares.

#### 5.4.1 Considerações finais

Os testes de consumo energético também foram utilizados para avaliar o impacto de diferentes limiares da MXC na entrega de pacotes.

Figura 5.53 – Latência com diferentes limiares de MXC na topologia 25D150.



Fonte: do autor (2017).

Como o baixo tráfego gerado pelos nós não foi capaz de estressar a capacidade de transmissão do Gen3, o único limiar que apresentou alguma variação foi o de 1%. Com esse valor ocorreu maior consumo energético nas topologias 25D25 e 25D150, bem como redução da latência. As métricas de vazão e taxa de entrega não tiveram variação em nenhum dos casos, sendo descartadas da análise.

Quanto ao consumo energético, cujos valores foram baseados no transceptor nRF24I01+, o Gen3 apresentou consumo energético consideravelmente menor que o MC-LMAC na grande maioria dos casos. Avaliando os resultados de consumo em conjunto com os resultados de entrega, pode-se afirmar que o Gen3 é um protocolo energeticamente mais eficiente que o MC-LMAC, dado que mostra maior capacidade de entrega de pacotes enquanto mantém um menor consumo médio de energia.

## 6 CONCLUSÕES

Este trabalho apresentou o Gen3, um protocolo MAC TDMA generalista que organiza a rede como uma árvore. A introdução do trabalho traz uma contextualização sobre RSSF e protocolos MAC, seguida pela motivação para o desenvolvimento do protocolo. O problema tratado pelo trabalho foi definido em detalhes, bem como a solução proposta para a solução deste problema.

Em seguida foram apresentados os principais conceitos que fundamentaram o desenvolvimento deste trabalho. Os conceitos básicos sobre RSSF, as arquiteturas de comunicação utilizadas nessas redes, e finalmente, discutidas as questões específicas relacionadas ao controle de acesso ao meio.

Por ser proposto para uso em redes críticas, o Gen 3 baseou-se em TDMA, pois esse tipo de protocolo tem comportamento mais previsível que protocolos CBP em redes densas ou com alto tráfego. Foram apresentadas diferentes formas de organização utilizadas nos protocolos TDMA, que dependem da sincronização de tempo dos nós. O problema de sincronização em RSSF foi analisado, e os resultados de alguns trabalhos fundamentaram decisões de projeto e de avaliação do Gen3, como o tamanho dos *slots*, a duração de tempos de guarda e os campos do pacote de controle.

O uso de múltiplos canais permite o aumento da capacidade de transmissão de dados dos protocolos MAC. Foram analisadas diversas questões pertinentes ao uso dessa técnica, incluindo os problemas de coordenação encontrados. Essas informações também serviram de base para a definição de mecanismos do Gen3 como: a lista de canais, que permite a definição de canais mais distantes evitando interferências de canais adjacentes; definição de tempos de guarda que incorporassem o tempo necessário para configuração de um novo canal no transceptor. Ainda visando aumentar a capacidade de transmissão do protocolo, o mecanismo de multiplexação estatística coordenada foi analisado. Apesar desse mecanismo ser utilizado em protocolos de telefonia há décadas, não foram encontrados trabalhos que o aplicassem em protocolos MAC para RSSF.

Dos diversos protocolos MAC analisados, nenhum combinou as características encontradas no Gen3: configuração automática, independente da densidade da rede e da movimentação de nós; uso de multicanal com um esquema simplificado de alocação de canais e suporte à introdução de mecanismos de cognição; uso de multiplexação estatística para o reaproveitamento de *slots* de tempo subutilizados. Essa avaliação fundamenta a proposta de desenvol-



vimento de um novo protocolo MAC, combinando esses mecanismos para oferecer suporte a aplicações que dependam de comunicação confiável, alta vazão de informações com baixa latência.

Os conceitos e a operação do Gen3 foram descritos em detalhes, incluindo os principais procedimentos da rede: associação e desassociação de nós; alocação de *frames*, slots e canais; e transmissão de pacotes com MXC ativada e desativada. A metodologia de avaliação do Gen3 combinou diversas topologias físicas para as redes, com variadas densidades e disposições dos nós, além de padrões de tráfego e mobilidade dos nós.

O Gen3 foi comparado ao MC-LMAC em termos de capacidade de transmissão de pacotes, considerando as métricas de vazão, taxa de entrega, latência e consumo energético. O Gen3 atingiu taxas de entrega e capacidade de vazão superiores ao MC-LMAC em todos os casos, mesmo com menor consumo energético. Em redes com baixa densidade foi possível configurar o MC-LMAC com menos *slots*, o que permitiu a obtenção de latências menores que o Gen3. No entanto, devido ao maior descarte de pacotes, esse ganho de latência pode ser anulado caso implique em retransmissões de pacotes pelas camadas superiores.

Para avaliar o impacto da mobilidade dos nós na operação do Gen3, foram realizados dois tipos de testes sobre a mesma rede. No primeiro teste, utilizado como parâmetro, os nós dessa rede não possuíam mobilidade, enquanto no segundo teste a mobilidade dos nós foi habilitada, permitindo avaliar isoladamente o impacto da mobilidade.

A adaptabilidade do Gen3 evidencia sua natureza generalista, permitindo a correta configuração da rede e alta capacidade de transmissão de dados sem que haja conhecimento prévio sobre a disposição dos nós. O uso da MXC se mostrou viável em protocolos para RSSF, resultando em ganho de performance no Gen3 em topologias não lineares, principalmente com a rede submetida a tráfego mais intenso. Em geral, o Gen3 se apresenta como um protocolo adequado a redes que executem operações que requeiram alto desempenho do protocolo MAC em termos de taxa de entrega, vazão de dados, latência, eficiência energética e suporte a mobilidade.

## 6.1 Trabalhos futuros

As seguintes atividades são definidas como trabalhos futuros, como forma de correção e melhoria do protocolo.

- Implementação e avaliação do Gen3 em *hardware* real. Todas as definições do protocolo levaram em consideração fatores e limitações encontradas em dispositivos de *hardware*

real. Os tempos de guarda definidos foram baseados em resultados apresentados por outros autores, cujos resultados provêm de testes em dispositivos reais. Ademais, os pacotes de controle e de dados têm formatos bem definidos e de tamanho fixo, que permitem à rede escalabilidade acima da apresentada nos testes.

- Modificação do formato de endereçamento. Em uma rede real, o formato de endereçamento dinâmico apresentado neste trabalho implicaria em um *overhead* adicional, com cada nó tendo 3 endereços: IP, endereço de *hardware* (endereço MAC), e endereço do protocolo Gen3. A proposta de modificação é que seja utilizado apenas o endereço MAC, e que o protocolo armazene em uma tabela os endereços MAC de seus nós filhos e de seu nó pai. O roteamento de pacotes seria tratado por um protocolo na camada de rede.
- Implementação de métodos de rádios cognitivos para a detecção de canais com alta interferência e sua substituição. O protocolo já define limites na lista de canais para que seja feita a negociação e troca de canais em caso de detecção de interferências. Com rádios cognitivos é possível aumentar a robustez da rede ou até identificar quando a quantidade de canais utilizada pode ser considerada baixa para a densidade da rede.
- Aumento da capacidade do *sink*. No atual formato, o *sink* opera na metade de sua capacidade total. No *slot 0* ele se comunica com seus nós filhos e no *slot 1* sempre fica ocioso. Nesse *slot* o *sink* pode permitir também a associação de nós, gerando duas redes paralelas e com o mesmo nó *sink*.
- Controle de admissão no protocolo de acordo com requisitos da aplicação. Com o crescimento da quantidade de nós associados, o tamanho do ciclo também aumenta, impactando na capacidade de entrega de pacotes e aumentando a latência. O protocolo pode empregar algum método para determinar que não deve permitir a associação de novos nós, de modo que para os nós já associados, seja possível manter uma determinada qualidade de serviço demandada pelas aplicações.

## REFERÊNCIAS

- ABDELI, D.; ZELIT, S.; MOUSSAOUI, S. Rth-mac: A real time hybrid mac protocol for wsn. In: **Programming and Systems (ISPS), 2013 11th International Symposium on**. [S.l.: s.n.], 2013. p. 153–162.
- AKYILDIZ, I. et al. Wireless Sensor Networks: A Survey. **Computer Networks**, v. 38, n. 4, p. 393 – 422, 2002. ISSN 1389-1286. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1389128601003024>>.
- AKYILDIZ, I. F.; KASIMOGLU, I. H. Wireless sensor and actor networks: research challenges. **Ad Hoc Networks**, v. 2, n. 4, p. 351 – 367, 2004. ISSN 1570-8705. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1570870504000319>>.
- AKYILDIZ, I. F. et al. A survey on spectrum management in cognitive radio networks. **IEEE Communications Magazine**, v. 46, n. 4, p. 40–48, April 2008. ISSN 0163-6804.
- AKYILDIZ, I. F.; MELODIA, T.; CHOWDHURY, K. R. A survey on wireless multimedia sensor networks. **Computer Networks**, v. 51, n. 4, p. 921 – 960, 2007. ISSN 1389-1286. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1389128606002751>>.
- ALI, M.; SULEMAN, T.; UZMI, Z. Mmac: a mobility-adaptive, collision-free mac protocol for wireless sensor networks. In: **Performance, Computing, and Communications Conference, 2005. IPCCC 2005. 24th IEEE International**. [S.l.: s.n.], 2005. p. 401–407. ISSN 1097-2641.
- BACHIR, A. et al. Mac essentials for wireless sensor networks. **Communications Surveys Tutorials, IEEE**, v. 12, n. 2, p. 222–248, Second 2010. ISSN 1553-877X.
- CHANDRA, K. Statistical multiplexing. **Encyclopedia of Telecommunications**, Wiley Online Library, 2003.
- CHINTALAPUDI, K. K. i-mac - a mac that learns. In: **Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks**. New York, NY, USA: ACM, 2010. (IPSN '10), p. 315–326. ISBN 978-1-60558-988-6. Disponível em: <<http://doi.acm.org/10.1145/1791212.1791250>>.
- CORREIA, L. b. et al. Dynmac: A resistant mac protocol to coexistence in wireless sensor networks. **Computer Networks**, v. 76, p. 1–16, 2015. Cited By 0. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-84912119349&partnerID=40&md5=c8edd438b2e4b2c9435e14957cb81339>>.
- DAY, J. The (un) revised osi reference model. **ACM SIGCOMM Computer Communication Review**, ACM, v. 25, n. 5, p. 39–55, 1995.
- DAY, J. D.; ZIMMERMANN, H. The osi reference model. **Proceedings of the IEEE**, IEEE, v. 71, n. 12, p. 1334–1340, 1983.
- DIAB, R.; CHALHOUB, G.; MISSON, M. Hybrid multi-channel mac protocol for wireless sensor networks: Interference rate evaluation. In: **Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th**. [S.l.: s.n.], 2013. p. 1–6. ISSN 1090-3038.
- DUNLOP, J. et al. Performance of a statistically multiplexed access mechanism for a tdma radio interface. **IEEE Personal Communications**, v. 2, n. 3, p. 56–64, Jun 1995. ISSN 1070-9916.

- ELSON, J.; GIROD, L.; ESTRIN, D. Fine-grained network time synchronization using reference broadcasts. **SIGOPS Oper. Syst. Rev.**, ACM, New York, NY, USA, v. 36, n. SI, p. 147–163, dez. 2002. ISSN 0163-5980. Disponível em: <<http://doi.acm.org/10.1145/844128.844143>>.
- FAIRHURST, G.; WOOD, L. **Advice to Link Designers on Link Automatic Repeat reQuest (ARQ)**. United States: RFC Editor, 2002.
- GREUNEN, J. van; RABAEY, J. Lightweight time synchronization for sensor networks. In: **Proceedings of the 2Nd ACM International Conference on Wireless Sensor Networks and Applications**. New York, NY, USA: ACM, 2003. (WSNA '03), p. 11–19. ISBN 1-58113-764-8. Disponível em: <<http://doi.acm.org/10.1145/941350.941353>>.
- HOESEL, L. F. V.; HAVINGA, P. J. A lightweight medium access protocol (lmac) for wireless sensor networks: Reducing preamble transmissions and transceiver state switches. Society of Instrument and Control Engineers (SICE), 2004.
- HUANG, P. et al. The evolution of mac protocols in wireless sensor networks: A survey. **Communications Surveys Tutorials, IEEE**, v. 15, n. 1, p. 101–120, First 2013. ISSN 1553-877X.
- INCEL, O. A survey on multi-channel communication in wireless sensor networks. **Computer Networks**, v. 55, n. 13, p. 3081–3099, 2011. Cited By 28. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-79961023720&partnerID=40&md5=c4f6def588e49fa1e4a12a457e497197>>.
- INCEL, O. D. et al. Multi-channel interference measurements for wireless sensor networks. In: **Proceedings. 2006 31st IEEE Conference on Local Computer Networks**. [S.l.: s.n.], 2006. p. 694–701. ISSN 0742-1303.
- INCEL, O. D. et al. Mc-lmac: A multi-channel mac protocol for wireless sensor networks. **Ad Hoc Networks**, v. 9, n. 1, p. 73 – 94, 2011. ISSN 1570-8705. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1570870510000624>>.
- INSTRUMENTS, T. CC2420: 2.4 GHz IEEE 802.15. 4/ZigBee-ready RF Transceiver. **Available at Available at <http://www.ti.com/lit/gpn/cc2420>**, p. 53, 2006.
- JANG, B.; LIM, J. B.; SICHITIU, M. L. As-mac: An asynchronous scheduled mac protocol for wireless sensor networks. In: **2008 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems**. [S.l.: s.n.], 2008. p. 434–441. ISSN 2155-6806.
- JHUMKA, A.; KULKARNI, S. On the design of mobility-tolerant tdma-based media access control (mac) protocol for mobile sensor networks. In: JANOWSKI, T.; MOHANTY, H. (Ed.). **Distributed Computing and Internet Technology**. Springer Berlin Heidelberg, 2007, (Lecture Notes in Computer Science, v. 4882). p. 42–53. ISBN 978-3-540-77112-8. Disponível em: <[http://dx.doi.org/10.1007/978-3-540-77115-9\\_4](http://dx.doi.org/10.1007/978-3-540-77115-9_4)>.
- JÚNIOR, G. M.; MARQUES, A. F. F.; CORREIA, L. H. A. Lmt-mac: um protocolo mac multicanal livre de colisões para redes de sensores sem fio. **Anais do XXXIV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**, v. 1, n. 1, p. 732–745, 6 2016.

- KARL, H.; WILLIG, A. **Protocols and Architectures for Wireless Sensor Networks**. [S.l.]: John Wiley & Sons, 2005. ISBN 0470095105.
- KUROSE, J.; ROSS, K. **Computer Networking: A Top-Down Approach**. [S.l.]: Pearson, 2013.
- LAM, S. S. A carrier sense multiple access protocol for local networks. **Computer Networks (1976)**, Elsevier, v. 4, n. 1, p. 21–32, 1980.
- LEE, H.; UN, C. Performance analysis of statistical voice/data multiplexing systems with voice storage. **IEEE Transactions on Communications**, v. 33, n. 8, p. 809–819, Aug 1985. ISSN 0090-6778.
- LIM, J. B.; JANG, B.; SICHITIU, M. L. Mcas-mac: A multichannel asynchronous scheduled {MAC} protocol for wireless sensor networks. **Computer Communications**, v. 56, p. 98 – 107, 2015. ISSN 0140-3664. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0140366414003375>>.
- MARÓTI, M. et al. The flooding time synchronization protocol. In: **Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems**. New York, NY, USA: ACM, 2004. (SenSys '04), p. 39–49. ISBN 1-58113-879-2. Disponível em: <<http://doi.acm.org/10.1145/1031495.1031501>>.
- MARQUES, A. F. F. et al. Iskra - an intelligent sensing protocol for cognitive radio. In: **2016 IEEE Symposium on Computers and Communication (ISCC)**. [S.l.: s.n.], 2016. p. 385–390.
- NORDIC\_SEMICONDUCTOR. **nRF24101 Product Specification v2.0**. [S.l.], 2013.
- O'DONOVAN, T. et al. The ginseng system for wireless monitoring and control: Design and deployment experiences. **ACM Trans. Sen. Netw.**, ACM, New York, NY, USA, v. 10, n. 1, p. 4:1–4:40, dez. 2013. ISSN 1550-4859. Disponível em: <<http://doi.acm.org/10.1145/2529975>>.
- PETERSEN, S.; CARLSEN, S. Wirelesshart versus isa100. 11a: The format war hits the factory floor. **Industrial Electronics Magazine, IEEE**, IEEE, v. 5, n. 4, p. 23–34, 2011.
- REINER, L.; FAIRHURST, G.; WOOD, L. **Advice for Internet Subnetwork Designers**. RFC Editor, 2004. RFC 3819. (Request for Comments, 3819). Disponível em: <<https://rfc-editor.org/rfc/rfc3819.txt>>.
- RHEE, I. et al. Z-mac: A hybrid mac for wireless sensor networks. **Networking, IEEE/ACM Transactions on**, v. 16, n. 3, p. 511–524, June 2008. ISSN 1063-6692.
- ROBERGE, C.; ADOUL, J. P. Fast on-line speech/voiceband - data discrimination for statistical multiplexing of data with telephone conversations. **IEEE Transactions on Communications**, v. 34, n. 8, p. 744–751, August 1986. ISSN 0090-6778.
- RODENAS-HERRAIZ, D. et al. Current trends in wireless mesh sensor networks: A review of competing approaches. **Sensors (Switzerland)**, v. 13, n. 5, p. 5958–5995, 2013. Cited By 5. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-84878103518&partnerID=40&md5=128a8cab5441c2c0b6734ba1a6617b15>>.

ROMER, K.; MATTERN, F. The design space of wireless sensor networks. **Wireless Communications, IEEE**, v. 11, n. 6, p. 54–61, Dec 2004. ISSN 1536-1284.

RUIZ, L. B. et al. Arquiteturas para Redes de Sensores Sem Fio. In: **22º Simpósio Brasileiro de Redes de Computadores**. [S.l.: s.n.], 2004. p. 167 – 218.

SHEN, W. et al. Prioritymac: A priority-enhanced mac protocol for critical traffic in industrial wireless sensor and actuator networks. **Industrial Informatics, IEEE Transactions on**, v. 10, n. 1, p. 824–835, Feb 2014. ISSN 1551-3203.

SO, J.; VAIDYA, N. H. Multi-channel mac for ad hoc networks: handling multi-channel hidden terminals using a single transceiver. In: ACM. **Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing**. [S.l.], 2004. p. 222–233.

SOMAPPA, A. K.; OVSTHUS, K.; KRISTENSEN, L. An industrial perspective on wireless sensor networks - a survey of requirements, protocols, and challenges. **Communications Surveys Tutorials, IEEE**, v. 16, n. 3, p. 1391–1412, Third 2014. ISSN 1553-877X.

SOMMER, P.; WATTENHOFER, R. Gradient clock synchronization in wireless sensor networks. In: **Proceedings of the 2009 International Conference on Information Processing in Sensor Networks**. Washington, DC, USA: IEEE Computer Society, 2009. (IPSN '09), p. 37–48. ISBN 978-1-4244-5108-1. Disponível em: <<http://dl.acm.org/citation.cfm?id=1602165.1602171>>.

SONG, J. et al. WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control. In: **Real-Time and Embedded Technology and Applications Symposium, 2008. RTAS '08. IEEE**. [S.l.: s.n.], 2008. p. 377–386. ISSN 1545-3421.

SONG, W.-Z. et al. Treemac: Localized tdma mac protocol for real-time high-data-rate sensor networks. **Pervasive and Mobile Computing**, v. 5, n. 6, p. 750 – 765, 2009. ISSN 1574-1192. PerCom 2009. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1574119209000649>>.

SOUA, R.; MINET, P. Multichannel assignment protocols in wireless sensor networks: A comprehensive survey. **Pervasive and Mobile Computing**, v. 16, n. PA, p. 2–21, 2015. Cited By 0. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-84921068075&partnerID=40&md5=03c1c8dbc0792458e574dccbe3f03536>>.

SURIYACHAI, P.; BROWN, J.; ROEDIG, U. Time-critical data delivery in wireless sensor networks. In: RAJARAMAN, R. et al. (Ed.). **Distributed Computing in Sensor Systems**. [S.l.]: Springer Berlin Heidelberg, 2010, (Lecture Notes in Computer Science, v. 6131). p. 216–229. ISBN 978-3-642-13650-4.

SURIYACHAI, P.; ROEDIG, U.; SCOTT, A. A survey of mac protocols for mission-critical applications in wireless sensor networks. **Communications Surveys Tutorials, IEEE**, v. 14, n. 2, p. 240–264, Second 2012. ISSN 1553-877X.

TANENBAUM, A. **Computer Networks**. 4th. ed. [S.l.]: Prentice Hall Professional Technical Reference, 2002. ISBN 0130661023.

TANENBAUM, A. S.; WETHERALL, D. J. **Computer Networks**. 5th. ed. [S.l.]: Prentice Hall, 2011. ISBN 978-0-13-212695-3.

VARGA, A. et al. The omnet++ discrete event simulation system. In: SN. **Proceedings of the European simulation multiconference (ESM'2001)**. [S.l.], 2001. v. 9, n. S 185, p. 65.

WU, Y. C.; CHAUDHARI, Q.; SERPEDIN, E. Clock synchronization of wireless sensor networks. **IEEE Signal Processing Magazine**, v. 28, n. 1, p. 124–138, Jan 2011. ISSN 1053-5888.

YU, G.-J.; CHANG, C.-Y. An efficient cluster-based multi-channel management protocol for wireless ad hoc networks. **Computer Communications**, v. 30, n. 8, p. 1742 – 1753, 2007. ISSN 0140-3664. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0140366407000643>>.

ZHIYONG, T.; DARGIE, W. A mobility-aware medium access control protocol for wireless sensor networks. In: **GLOBECOM Workshops (GC Wkshps), 2010 IEEE**. [S.l.: s.n.], 2010. p. 109–114.

ZHOU, G. et al. Mmsn: Multi-frequency media access control for wireless sensor networks. In: **Infocom**. [S.l.: s.n.], 2006. v. 6, p. 1–13.