



**PROTOCOLO MAC EFICIENTE EM ENERGIA
E LIVRE DE COLISÕES PARA
REDES DE SENSORES SEM FIO**

Vinícius Galvão Guimarães

**DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia
Departamento de Engenharia Elétrica

**PROTOCOLO MAC EFICIENTE EM ENERGIA
E LIVRE DE COLISÕES PARA
REDES DE SENSORES SEM FIO**

Vinícius Galvão Guimarães

Orientador: Renato Mariz de Moraes

Coorientador: Adolfo Bauchspiess

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA

Publicação: PPGEA.DM - 573/14

Brasília, Agosto de 2014

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA

**PROTOCOLO MAC EFICIENTE EM ENERGIA
E LIVRE DE COLISÕES PARA
REDES DE SENSORES SEM FIO**

Vinícius Galvão Guimarães

*Relatório submetido ao Departamento de Engenharia
Elétrica como requisito parcial para obtenção
do grau de Mestre Engenheiro em Sistemas Eletrônicos e Automação*

Banca Examinadora

Prof. Renato Mariz de Moraes, UnB, Brasil.

Orientador

Prof.^a Kátia Obraczka, UCSC, Santa Cruz, CA, USA

Examinadora externa

Prof. Paulo Roberto de Lira Gondim, UnB, Brasil

Examinador externo

FICHA CATALOGRÁFICA

GUIMARÃES, VINÍCIUS GALVÃO

Protocolo MAC Eficiente em Energia e Livre de Colisões para Redes de Sensores Sem Fio,
[Distrito Federal] 2014.

xiii, 87p., 210 x 297 mm (ENE/FT/UnB, Mestre, Engenharia Elétrica, 2014). Dissertação de Mestrado – Universidade de Brasília. Faculdade de Tecnologia.

1. Livre de colisão

2. Eficiência energética

3. Protocolo MAC

4. Temporização dinâmica

I. ENE/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

GUIMARÃES, V. G., (2014). Protocolo MAC Eficiente em Energia e Livre de Colisões para Redes de Sensores Sem Fio. Dissertação de Mestrado em Engenharia Elétrica, Publicação PPGEA.DM - 573/14, Departamento de Engenharia Elétrica, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 87p.

CESSÃO DE DIREITOS

AUTOR: Vinícius Galvão Guimarães

TÍTULO DA DISSERTAÇÃO DE MESTRADO: Protocolo MAC Eficiente em Energia e Livre de Colisões para Redes de Sensores Sem Fio.

GRAU: Mestre

ANO: 2014

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor.

Vinícius Galvão Guimarães

Departamento de Eng. Elétrica (ENE) - FT

Universidade de Brasília (UnB)

Campus Darcy Ribeiro

CEP 70919-970 - Brasília - DF - Brasil

Dedicatória

Aos meus pais, por sempre me guiarem pelos caminhos, acreditarem em meu potencial e me darem força para seguir em frente. À minha companheira de jornada, pela credibilidade, apoio e paciência. À minha irmã que é e sempre será minha amiga e companheira. Aos meus familiares, que sempre torceram pelo meu sucesso e acompanharam minha trajetória, em especial meus avós que me guiam junto à Deus. Aos meus amigos, pela força. A Deus, por tudo que tenho e sou.

Vinícius Galvão Guimarães

Agradecimentos

Agradeço principalmente aos meus pais, que sempre estiveram presentes, nos momentos de felicidade e tristeza, me levantando sempre que precisei.

Ao meu orientador e coorientador que acreditaram em meu trabalho e compartilharam seus conhecimentos para a realização deste.

A todos meus amigos e colegas que estiveram presentes nessa jornada e fizeram dela uma excelente viagem, engrandecedora e inestimável.

A todos os colaboradores e funcionários que me ajudaram e possibilitaram chegar onde estou.

Ao laboratório LARA e seus alunos que me apoiaram e fazem dele um ambiente ideal para se trabalhar. Um lugar pequeno em espaço, mas um dos grandes centros de inteligência e qualidade técnica da UnB.

À Faculdade de Tecnologia e à Universidade de Brasília por me dar essa oportunidade gloriosa de levar estes nomes em meu currículo.

À CAPES pelo apoio financeiro nesta pesquisa.

Vinícius Galvão Guimarães

RESUMO

Este trabalho propõe e desenvolve o DyTEE (do inglês *Dynamic Timed Energy Efficient*), um novo protocolo de controle de acesso ao meio (MAC) livre de colisões de dados para redes de sensores sem fio, apropriado principalmente a ambientes de automação predial e rural. Este foi baseado em Múltiplo Acesso por Divisão do Tempo (TDMA) com ciclos de ativação e períodos de transmissão dinâmicas, empregando somente a utilização de um canal e sensoriamento da portadora.

O novo protocolo foi implementado em módulos reais na plataforma Zigbit da Atmel, utilizando o pacote “IEEE 802.15.4 MAC” fornecido pelo fabricante. Foi também implementado em simulação, utilizando parâmetros fornecidos pelo autor do S-MAC (plataforma MICA e sistema operacional TinyOS) e os parâmetros obtidos da implementação em Zigbit. Dessa forma, o DyTEE foi comparado com o S-MAC e o IEEE 802.15.4, onde foram aferidas métricas de vazão, latência, consumo energético e outras.

Os resultados dos experimentos sob diferentes parâmetros e topologias mostraram que a nova proposta supera ambos os protocolos comparados. Isto é, em relação ao S-MAC, apresentou aproximadamente um quinto do consumo de energia, bem como melhor latência e vazão aproximadamente até 80% maior. Enquanto que em relação ao IEEE 802.15.4, apresentou também melhores resultados, principalmente em relação ao tempo de vida estimado do módulo real realizando uma tarefa de sensoriamento de temperatura, no qual o DyTEE obteve um ganho de aproximadamente 25,6% a mais de duração do que o IEEE 802.15.4.

Concluiu-se que o novo protocolo possui uma boa economia de energia, mantendo bons resultados de latência e vazão, bem como este foi implementado em módulos de baixo poder computacional e restrita memória, ou seja, um protocolo compacto e eficiente.

Palavras Chave: Livre de colisão, Eficiência energética, Protocolo MAC, Temporização dinâmica.

ABSTRACT

This paper proposes develops the DyTEE (Dynamic Energy Efficient Timed), a new protocol for medium access control (MAC) collision-free data for wireless sensor networks, designed for building automation and rural automation environments. It was based on Time Division Multiple Access (TDMA) with activation cycles and dynamics transmission slots, employing the use of only one channel and carrier sensing.

The new protocol has been implemented in real modules Zigbit Atmel platform, using the "IEEE 802.15.4 MAC" package provided by the manufacturer. Was also implemented in simulation using parameters provided by the author of the S-MAC (MICA platform and TinyOS operating system) and the parameters obtained in the implementation with Zigbit. Thus, DyTEE was compared with S-MAC and IEEE 802.15.4, which were measured metrics of throughput, latency, energy consumption and other.

The results of experiments under different topologies and parameters have shown that the new proposal outperforms both protocols compared. That is, relative to the S-MAC, DyTEE showed about one fifths of the energy consumption as well as improved latency and throughput with approximately 80% greater. While compared to the IEEE 802.15.4, DyTEE also showed better results, especially regarding the estimated life time of the actual module performing a task of sensing temperature, in which the DyTEE obtained a gain of approximately 25.6% more duration than the IEEE 802.15.4.

It was concluded that the new protocol has a good energy saving, while maintaining good throughput and latency results, as well it was implemented in low computational power and limited memory modules, that means, DyTEE is a compact and efficient protocol.

Keywords: Collision free, Energy Efficiency, MAC Protocol, Dynamic timed.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	MOTIVAÇÃO	1
1.2	REDE DE SENSORES E ATUADORES	2
1.3	DESCRIÇÃO DO PROBLEMA	4
1.4	OBJETIVO DO TRABALHO	5
1.5	APRESENTAÇÃO DO MANUSCRITO	6
2	TRABALHOS RELACIONADOS	7
2.1	PROTOCOLOS DE COMUNICAÇÃO	9
2.1.1	SENSOR MAC	9
2.1.2	SENSOR MAC COM ESCUTA ADAPTATIVA	10
2.1.3	MÚLTIPLO ACESSO POR DIVISÃO DO TEMPO (TDMA)	11
2.1.4	SENSORIAMENTO DA PORTADORA E MÚLTIPLO ACESSO COM PREVENÇÃO DE COLISÃO (CSMA/CA)	12
2.1.5	PROTOCOLO IEEE 802.15.4	12
2.1.6	T-MAC	14
2.1.7	WISEMAC	15
2.1.8	B-MAC	16
2.1.9	X-MAC	17
2.1.10	AEDP E OUTROS PROTOCOLOS COM BASE EM LPL	18
2.1.11	DMAC	19
2.1.12	TDMA-WAKEUP	20
2.1.13	ID-MAC	20
2.1.14	AS-MAC	21
2.2	TOPOLOGIAS UTILIZADAS	22
2.2.1	LINEAR	23
2.2.2	ESTRELA	23
2.3	QUADRO COMPARATIVO	24
2.4	CONCLUSÃO DO CAPÍTULO	26
3	PROTOCOLO <i>Dynamic Timed Energy Efficient</i> (DYTEE)	27
3.1	PARÂMETROS CONFIGURÁVEIS DO DYTEE	28
3.2	INÍCIO DE UMA REDE	31

3.3	ESTRUTURA GERAL DO PROTOCOLO.....	31
3.4	ESTRUTURA DO MOMENTO DE SINCRONIZAÇÃO.....	34
3.4.1	SINAL DE SINCRONIZAÇÃO - BS (<i>Beacon Signal</i>)	34
3.4.2	PERÍODO DE REGISTRO DE NOVOS NÓS - REG	35
3.4.3	REQUISIÇÃO DE VAGAS DE TRANSMISSÃO - R SLOT	37
3.4.4	DIVULGAÇÃO DA AGENDA - <i>Broadcast</i>	37
3.4.5	PARÂMETROS DEPENDENTES.....	39
3.5	TRANSMISSÃO DE PACOTES DE DADOS	40
3.6	DESCRIÇÃO DO PROTOCOLO DYTEE PARA UM NÓ DO GRUPO	42
3.7	CONCLUSÃO DO CAPÍTULO	42
4	METODOLOGIA	44
4.1	DESENVOLVIMENTO E SIMULAÇÕES	44
4.2	ESCOLHA DA PLATAFORMA DE IMPLEMENTAÇÃO	45
4.3	DESENVOLVIMENTO EM MÓDULO REAL.....	46
4.4	OBTENÇÃO DE RESULTADOS.....	48
4.4.1	MÉTODO DE CONTAGEM DE BYTES TRANSMITIDOS E RECEBIDOS	48
4.4.2	MÉTODO DE MEDIÇÃO EM MULTÍMETRO DIGITAL	51
4.5	CONCLUSÃO DO CAPÍTULO	53
5	RESULTADOS E DISCUSSÕES	55
5.1	SIMULAÇÕES COM PARÂMETROS S-MAC	55
5.2	SIMULAÇÕES COM PARÂMETROS ATUALIZADOS	61
5.3	RESULTADOS DO DYTEE IMPLEMENTADO EM MÓDULOS REAIS	65
5.3.1	MÉTODO DE CONTAGEM DE BYTES TRANSMITIDOS E RECEBIDOS	65
5.3.2	MÉTODO DE MEDIÇÃO EM MULTÍMETRO DIGITAL	68
5.4	COMPARAÇÃO DO DYTEE E O IEEE 802.15.4 EM MÓDULOS REAIS	72
5.5	TEMPO DE VIDA DO MÓDULO REAL	76
5.6	CONCLUSÃO DO CAPÍTULO	77
6	CONCLUSÕES E PERSPECTIVAS	78
6.1	CONCLUSÕES FINAIS.....	78
6.2	PERSPECTIVAS.....	79
	REFERÊNCIAS BIBLIOGRÁFICAS	80
	ANEXOS.....	85
I	DIAGRAMAS ESQUEMÁTICOS	86
I.1	PLACA DE INTEGRAÇÃO DO ZIGBIT - ESQUEMÁTICO	86
II	CIRCUITOS ELÉTRICOS EM PCB.....	87
II.1	PLACA DE INTEGRAÇÃO DO ZIGBIT - CIRCUITO	87

LISTA DE FIGURAS

1.1	Exemplo de rede de sensores enviando uma mensagem por uma rota específica.	3
1.2	Aplicações de rede de sensores e atuadores em: a) Ambientes inteligentes (Automação Predial) e b) Automação de irrigação na agricultura.....	4
2.1	Diagrama temporal de funcionamento do S-MAC.....	10
2.2	Diagrama temporal de funcionamento do S-MAC-AL.....	10
2.3	Diagrama temporal de funcionamento do TDMA.	11
2.4	Diagrama temporal de funcionamento do CSMA/CA.....	12
2.5	Diagrama temporal do <i>superframe</i> realizado no protocolo IEEE 802.15.4.	13
2.6	Diagrama temporal de funcionamento do T-MAC em comparação com S-MAC.	15
2.7	Diagrama temporal de funcionamento do WiseMAC.	16
2.8	Diagrama temporal comparativo entre o funcionamento do B-MAC e X-MAC.	18
2.9	Diagrama temporal e estrutura de roteamento do DMAC.	19
2.10	Topologia linear.	23
2.11	Topologia estrela.	24
3.1	Diagrama simplificado de eventos temporais do protocolo DyTEE.....	32
3.2	Diagrama temporal de um momento de sincronização (SyncM).....	34
3.3	Fluxograma de registro na fase REG do SyncM.	36
3.4	Fluxograma de operação para um nó que está executando o protocolo DyTEE.....	43
4.1	Módulo MICAZ.	45
4.2	Módulo Zigbit (modelo ATZB-24-A2).....	46
4.3	Módulo Zigbit integrado com PCB para acesso às saídas: a) com USB; b) Com bateria, sem USB.	47
4.4	Estrutura do pacote aberto IEEE 802.15.4 MAC disponibilizado pela Atmel.....	48
4.5	Processo de medição de corrente: a) Em um módulo somente; b) Em um módulo dentro da rede de sensores.....	51
4.6	Gráfico obtido do processo de medição de corrente de um nó da rede.	52
4.7	Gráfico obtido do processo de medição de corrente de um nó da rede, ampliado em um dos picos.	52
5.1	Total de energia consumida por todos os nós para enviar 20 pacotes de 100 bytes cada, da fonte para o nó de destino.	56

5.2	Série temporal da quantidade de nós ligados e recebendo por tempo em ambos protocolos.	57
5.3	Série temporal da quantidade de nós ligados e recebendo por tempo em ambos protocolos.	57
5.4	Série temporal da quantidade de nós ligados e transmitindo por tempo em ambos protocolos.	58
5.5	Consumo médio do nó líder comparado à média dos demais.	59
5.6	Média de latência de pacotes para o envio do nó fonte ao nó de destino.	59
5.7	Excesso de cabeçalho calculado no processo.	60
5.8	Vazão ao longo de toda a rede.	60
5.9	Total de energia consumida em ambos protocolos segundo diferentes tipos de chegada de mensagens.	62
5.10	Vazão de ambos protocolos segundo diferentes tipos de chegada de mensagens.	62
5.11	Excesso de cabeçalho de ambos protocolos segundo diferentes tipos de chegada de mensagens.	63
5.12	Latência de ambos protocolos segundo diferentes tipos de chegada de mensagens.	63
5.13	Consumo médio de um nó em ambos protocolos segundo diferentes tipos de chegada de mensagens.	64
5.14	Gráfico da combinação de três métricas relacionando ambos protocolos.	65
5.15	Vazão na rede simulada e real.	66
5.16	Consumo total de energia da rede simulado e real.	66
5.17	Consumo total de energia da rede em seus quatro estados: a) Transmitindo; b) Recebendo; c) Ocioso e d) Em dormência.	67
5.18	Excesso de cabeçalhos entre os experimentos reais e simulados.	67
5.19	Potência média consumida por um módulo operando o protocolo DyTEE nos procedimentos reais e simulados.	68
5.20	Comparação medição da corrente elétrica do protocolo DyTEE: a) Nos experimentos com falhas de implementação/plataforma; b) No padrão desejado.	68
5.21	Gráfico da medição de corrente elétrica do protocolo DyTEE de um módulo na rede apresentando erros de implementação/plataforma.	69
5.22	Gráfico de consumo total do protocolo DyTEE, comparado aos demais métodos de avaliação.	70
5.23	Gráfico da potência média consumida por um nó executando o protocolo DyTEE, comparado aos demais métodos de avaliação.	70
5.24	Gráfico de consumo total do protocolo DyTEE, sem erros de implementação/plataforma, comparado aos demais métodos de avaliação.	71
5.25	Gráfico de consumo total do protocolo DyTEE, sem erros de implementação/plataforma, comparado aos demais métodos de avaliação.	71
5.26	Total de energia consumida por todos os nós para enviar 50 pacotes de 20 bytes cada.	73
5.27	Vazão obtida em ambos protocolos em módulos reais.	73
5.28	Excesso de cabeçalho de ambos protocolos em módulos reais.	74
5.29	Potência média consumida por um módulo real em ambos protocolos.	74

5.30 Métrica combinada dupla para ambos protocolos em módulos reais.....	75
I.1 Esquemático da placa de integração do Zigbit.	86
II.1 Circuito da placa de integração do Zigbit visto: a) Somente marcações da frente; b) Somente marcações de trás; c) Circuito da frente e d) Circuito de trás.	87

LISTA DE TABELAS

2.1	Resultados do tamanho da implementação, em bytes, dos protocolos B-MAC e S-MAC em motes reais.....	17
2.2	Tabela comparativa entre os protocolos apresentados neste capítulo.....	25
3.1	Sequência de bytes do pacote de controle BS.....	34
3.2	Sequência de bytes do pacote de requisição de registro.....	36
3.3	Sequência de bytes do pacote de controle RSlot.....	37
3.4	Sequência de bytes do pacote de controle enviados no <i>broadcast</i>	38
3.5	Sequência de bytes do pacote de dados.....	41
3.6	Sequência de bytes do pacote de controle ACK.....	41
5.1	Parâmetros usados para simular os protocolos na topologia linear.....	56
5.2	Parâmetros usados para simular os protocolos na topologia estrela.....	61

LISTA DE SÍMBOLOS

Siglas e Acrônimos

MAC	Medium Access Control
QoS	Quality of Service
WSAN	Wireless Sensor and Actors Networks
TinyOS	TinyOS is a small Operational System
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
TDMA	Time Division Multiple Access
S-MAC	Sensor Medium Access Control
S-MAC-AL	Sensor Medium Access Control with Adaptive Listening
TRAMA	Traffic-Adaptive Medium Adaptive
BO	Beacon Order
SO	Superframe Order
CAP	Contention Access Period
GTS	Guaranteed Time Slots
CFP	Contention-Free Period
T-MAC	Timeout Medium Access Control
WiseMAC	Wise Medium Access Control
B-MAC	Berkeley Medium Access Control
AEDP	Adaptive Energy Detection Protocol
A-MAC	Auto-ack Medium Access Control
TDMA-W	Time Division Multiple Access - Wakeup
AS-MAC	Adaptive Spreading Medium Access Control
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
RTS	Request to Send
CTS	Clear to Send
ACK	Acknowledgment
LPL	Low Power Listening
ROM	Read Only Memory

Siglas e Acrônimos (continuação)

RAM	Random Access Memory
ID	Identification Number
GPS	Global Positioning System
SCADA	Supervisory Control and Data Acquisition
SyncM	Synchronization Moment
FCF	Frame Control Field
FCS	Frame Check Sequence
LQI	Link Quality Indication
ED	Energy Detection
PanID	Personal Area Network Identification
RTR	Request to Register
RACK	Register Acknowledgment
RSlot	Request Slot to Transmit
NR	Number of Registers
ND	Number of Disassociation
DSN	Data Sequence Number
PCB	Printed Circuit Board
USB	Universal Serial Bus
LARA	Laboratório de Automação e Robótica

Capítulo 1

Introdução

Este capítulo contextualiza a importância dos sistemas sem fio em ambientes de automação, o que motivou o desenvolvimento deste trabalho. Ao final são expostos alguns problemas atuais e os objetivos deste projeto.

1.1 Motivação

Na automação se faz necessário o uso de atuadores e sensores controlados à distância, e neste contexto surge a necessidade da comunicação entre dispositivos para que realizem determinada tarefa.

Considerando ambientes de produção, onde a distância entre os componentes geralmente não chega a quilômetros, a comunicação entre dispositivos foi, durante muito tempo, feita por cabos elétricos e protocolos de comunicação desenvolvidos considerando os problemas associados ao meio físico cabeado. Nestas condições, os sinais chegam quase instantaneamente e, dependendo das distâncias, com pouco ou sequer nenhum ruído. Nestes sistemas, a perda de informação na comunicação é muito pequena, isso sem considerar o uso de algoritmos de recuperação de dados.

Os sistemas cabeados foram e ainda são muito utilizados em diversas aplicações devido a seu custo, facilidade e acessibilidade, principalmente em ambientes em que a distância entre os componentes comunicantes é pequena. Entretanto a utilização de cabos exige uma certa estrutura para deixá-los instalados ou, no caso de uma nova instalação, pode exigir a realização de obras, reformas do ambiente e custos extras com outros serviços, sem contar a possível necessidade de dar voltas com os fios pelo ambiente, aumentando o custo com material e a poluição visual.

Existem também outras aplicações que não favorecem a utilização de cabos, como ambientes rurais, onde os cabos devem percorrer longas distâncias (aumento do custo por material) e propiciando, em casos de tempestades elétricas, a queima de componentes elétricos envolvidos na comunicação presente nestes cabos.

Com o avanço da tecnologia nestas últimas décadas, as comunicações sem fio foram ficando cada vez mais acessíveis e se tornando mais presente em diversos ambientes e sistemas. Nesta tecnologia

ocorrem peculiaridades não observadas nas comunicações a cabo, como: atrasos maiores entre mensagens, maior falha nas recepções de dados (mais interferências e ruídos), maior dificuldade de resolver problemas de dessincronização, problemas de terminais expostos, terminais escondidos, entre outras complicações.

Segundo [1], interferências por terminal escondido são causadas por transmissões simultâneas entre dois módulos que não estão ao alcance um do outro, mas estão ambos ao alcance de um terceiro módulo receptor destino das mensagens destes dois. Semelhantemente, as interferências por terminal exposto ocorrem quando um módulo é impedido de transmitir pacotes para outros nós devido a um módulo vizinho já em comunicação e transmitindo.

Dadas estas peculiaridades, os protocolos de comunicação e acesso ao meio (MAC - do inglês *Medium Access Control*) tiveram que ser adaptados para melhor tratarem estes problemas. Assim, o resultado obtido de estudos até hoje é bastante satisfatório, permitindo a inclusão desta tecnologia em diversas aplicações, mesmo em casos de alta exigência de confiabilidade e baixo atraso das mensagens. Contudo, para outros casos, devido à maior perda de pacotes relativamente à comunicação cabeada, a tecnologia a cabo ainda continuará sendo utilizada, bem como deverá possuir, na maioria dos casos, melhores resultados de velocidade, atraso e custo. Outra divergência entre essas formas de comunicação é o fato de que, diferentemente das comunicações cabeadas, o gasto energético com transmissões sem fio chega a ser 12 vezes maior do que o consumo do próprio processador no caso dos módulos Telos [2]. É importante citar que em relação aos meios de gerenciamento da energia dos módulos, a camada MAC possui grande importância [3].

Existem ainda situações em que se necessita monitorar e atuar em ambientes de automação em que cada módulo atuador/sensor é alimentado por uma bateria e sendo assim, a economia de energia se torna um dos principais parâmetros das aplicações. Contudo, sistemas de automação predial ou rural possuem certas características que permitem enfatizar a economia de energia sobre outros parâmetros envolvidos na qualidade de serviço (QoS - do inglês *Quality of Service*).

Nestas aplicações específicas, o intervalo entre cada mensagem, ou seja, a taxa de geração de mensagens é geralmente pequena, na casa de uma mensagem a cada alguns minutos, sendo que cada mensagem transmitida pode possuir poucos bytes de informação, uma vez que se tratam geralmente de comandos, leituras de temperatura e etc. Isso significa que a vazão destes sistemas pode ser baixa, juntamente com uma maior latência, sem qualquer prejuízo ao sistema. Utilizar destas características em prol da economia de energia é bastante desejável no desenvolvimento de um protocolo de comunicação específico para estes casos.

1.2 Rede de sensores e atuadores

Redes de sensores e atuadores (WSAN- do inglês *Wireless Sensor and Actor Networks*) é um tópico bem estudado atualmente e, com o avanço da tecnologia, estes sistemas serão ainda mais presentes na vida humana devido ao grande número de aplicações aos quais essas redes podem atender [4]. Dos antigos grandes circuitos integrados seguindo para pequenas cápsulas micrométricas, até a nano computação hoje em dia. Uma característica marcante nesta tecnologia

é a presença das baterias e a elevada dependência da energia consumida pelo módulo.

Por outro lado, o meio de armazenamento de energia não seguiu a miniaturização na mesma velocidade. Nesse sentido, mesmo que os componentes consumam menos energia, para se ter uma bateria menor será necessário reduzir a capacidade de armazenamento total desta. Isto produz uma demanda por meios da economia durante a operação dos nós em uma rede.

Na Figura 1.1, pode-se observar um exemplo do uso de módulos sensores e atuadores alimentados por bateria.

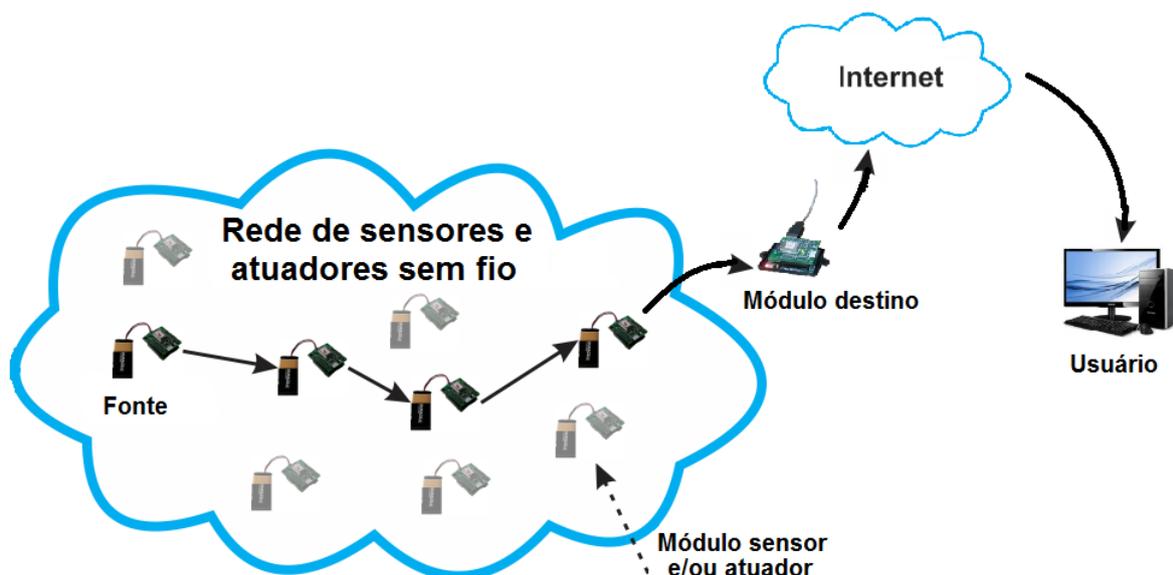


Figura 1.1: Exemplo de rede de sensores enviando uma mensagem por uma rota específica.

Neste exemplo da Figura 1.1, pode-se observar um nó (módulo sensor) gerador (fonte, do inglês *source*) de uma determinada mensagem enviando esta por meio de outros nós na rede até atingir seu destinatário (em inglês *sink*), onde este irá retransmitir através de outras interfaces para o sistema de monitoramento do usuário.

Existem ainda casos onde se deseja lançar vários sensores em uma determinada área, de forma que possam coletar os dados e retransmitir até chegar na central. Dessa forma, este sistema requer uma melhor eficiência energética para que o tempo de operação destes sensores seja viável à aplicação.

O grande benefício das redes de sensores é o fato de atender aos requisitos de aplicações citadas anteriormente, onde se deseja que todos os atuadores/sensores sejam capazes de se comunicar com uma central e ao mesmo tempo não se pode ou não é interessante utilizar cabos de comunicação e alimentação para cada um dos módulos. Dois exemplos disto são apresentados nas Figuras 1.2(a) e 1.2(b).

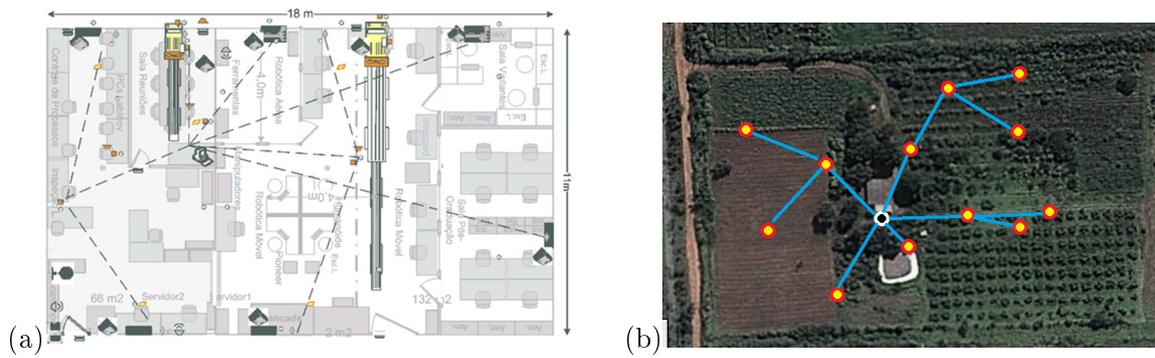


Figura 1.2: Aplicações de rede de sensores e atuadores em: a) Ambientes inteligentes (Automação Predial) e b) Automação de irrigação na agricultura.

No caso de um laboratório com ar condicionado e sensores de temperatura, o uso de WSNs torna o sistema muito mais eficiente e organizado, podendo haver o controle de vários aparelhos em vários ambientes do laboratório administrados por somente uma central sem cabos passando por todas as paredes ou cantos. O mesmo pode ser visto no exemplo da Figura 1.2(b), onde pode se evitar o uso de longos cabos, como inicialmente proposto no trabalho [5], e a possibilidade de avarias em casos de raios. Contudo, a dependência de baterias nestes exemplos torna o sistema vulnerável à falta de energia (bateria exaurida) de energia.

1.3 Descrição do problema

Procurando melhor atender às aplicações dos tipos apresentados nas Figuras 1.2(a) e 1.2(b), deseja-se focar o máximo possível em poupar energia, mantendo a QoS necessária para o funcionamento destes ambientes. Para isso, existem várias maneiras de economizar energia, tais como o uso de melhores materiais, melhores técnicas de programação para evitar computação desnecessária, permanecer em modo *sleep* (estado econômico) durante tempos de inatividade, entre outros. Adicionalmente, é possível também reduzir o consumo de energia por meio do protocolo de comunicação responsável pelo controle de acesso ao meio e camada física, pois neste pode haver um grande gasto de energia quando os pacotes transmitidos colidem e, portanto, não são capturados pelos receptores, bem como quando simplesmente um receptor fica continuamente ouvindo informações desnecessárias até que este receba um pacote destinado a ele. Assim, o projeto de protocolos na camada MAC focado em economia de energia é um tema importante para redes de sensores.

Tendo em vista estas necessidades, faz-se necessário o ajuste de certos parâmetros de qualidade de serviço da rede para melhor adequar o protocolo à aplicação. Como visto em [6], alguns dos principais parâmetros trabalhados na literatura para Rede de Sensores e Atuadores (WSAN) são:

- **Taxa de transferência (vazão ou do inglês *Throughput*):** Quantidade média de bytes de dados úteis por tempo enviados das fontes geradores de mensagem até o módulo de destino final;
- **Latência (do inglês *delay*):** Média de tempo para uma determinada mensagem viajar da

fonte para o destino final. Em controle, têm-se o tempo de atraso máximo tolerado, após este tempo considera-se o pacote perdido e um novo pacote é enviado ou solicitado;

- **Excesso de cabeçalho ou sobrecarga (do inglês *overhead*):** Quantidade média de bytes de controle (incluindo cabeçalhos, sincronizações e confirmações) por tempo para o envio da fonte ao destino final, ou seja, bytes de controle/bytes totais;
- **Consumo de energia:** Energia consumida da bateria que alimenta o nó sensor.

Entre muitas aplicações de redes de sensores sem fio, alguns de seus subsistemas como controle de temperatura em uma sala ou uma irrigação agrícola (como visto em [7]) podem possuir requisitos específicos. Mais detalhadamente:

- Não há grande vazão de dados, porque a geração destes não é muito frequente e geralmente com pacotes pequenos (menores que 100 bytes);
- Possibilidade de alta latência (em alguns casos, atrasos de minutos são toleráveis);
- Baixa densidade de nós;
- Dinâmica de conexão baixa, isso é, a tendência dos nós é de se conectarem na rede e permanecerem lá até esgotarem sua bateria. Dessa forma ocorre uma tendência de se trabalhar com roteamentos fixos;
- Necessidade de eficiência em energia para atingir maior longevidade, assim reduzir o número de trocas de baterias.

1.4 Objetivo do trabalho

Este trabalho desenvolve um novo protocolo MAC para redes de sensores e atuadores sem fio que não apresente colisões de dados, já que é baseado no método de múltiplo acesso e divisão do tempo (TDMA - do inglês *Time Division Multiple Access*) [8], possuindo ainda baixo consumo de energia e rendimento adequado à aplicação.

Para isso, foram realizados os seguintes objetivos específicos:

- Projetou-se e desenvolveu-se técnicas de comunicação para atender às especificações 1.3;
- Simulou-se o novo protocolo em uma plataforma de simulação, no caso, por meio de programação (*script*) em MATLAB [9] baseado em eventos;
- Comparou-se, em simulação, a nova proposta com o protocolo S-MAC [10], reconhecido protocolo MAC na literatura;
- Instalou-se o novo protocolo em uma plataforma real;
- Comparou-se nova proposta com o atual estado da arte e amplamente utilizado protocolo IEEE 802.15.4 [11].

1.5 Apresentação do manuscrito

O restante deste trabalho está estruturado da seguinte forma:

- O Capítulo 2 expõe os trabalhos relacionados, de forma a apresentar as principais características de protocolos de comunicação considerados relevantes. Considerando as aplicações de automação predial ou rural desejadas, foram analisados protocolos de diferentes métodos de operação, observando suas funções que melhor atendem a estas aplicações. Ao final do capítulo foram descritas topologias de rede que podem simular o ambiente destas aplicações, sendo que uma permite uma análise mais acadêmica (topologia linear), enquanto a outra trabalha de forma mais próxima dos ambientes de automação (topologia estrela).
- O Capítulo 3 descreve o novo protocolo DyTEE de forma detalhada, explicando cada processo necessário para seu funcionamento, bem como suas propriedades e características. Neste capítulo também podem ser vistos diagramas de operação do protocolo, a estrutura de seus pacotes e o fluxograma de procedimentos do protocolo sob o ponto de vista de um módulo da rede.
- O Capítulo 4 apresenta a metodologia utilizada para averiguar a eficiência do novo protocolo, descrevendo os motivos para a escolha dos componentes, plataformas e parâmetros. Neste capítulo expõe-se também as etapas dos procedimentos realizados, definições matemáticas das métricas utilizadas e estruturas do experimento.
- O Capítulo 5 apresenta o resultado com o desempenho do novo protocolo comparado ao S-MAC [10] e ao IEEE 802.15.4 [11]. Neste são detalhadas as comparações entre os resultados simulados e reais, bem como dados obtidos de simulações do protocolo S-MAC utilizando os parâmetros atualizados da nova plataforma real. Por fim, faz-se uma estimativa de tempo de vida de um módulo alimentado por bateria coletando dados de temperatura do ambiente e operando o DyTEE para se comunicar.
- Capítulo 6 discute as conclusões deste trabalho seguido de trabalhos futuros, referências e anexos.

Capítulo 2

Trabalhos relacionados

Este capítulo expõe e analisa trabalhos anteriores que estudam características interessantes que atenderiam as especificações para às aplicações específicas desejadas no presente estudo.

Devido ao grande número de aplicações cabíveis de uso de comunicações sem fio, atualmente existem muitos protocolos com características e estratégias para realizar diferentes tarefas. Entretanto, algumas dessas propostas consideram o uso de diferentes tecnologias ou mesmo o uso de mais de um só canal para seu funcionamento.

Segundo [12], protocolos da camada MAC podem ser classificados por aspectos que ajudam a decidir à qual aplicação este atenderá melhor. Assim, quatro divisões são muito importantes quanto ao modo de operação:

- **Nenhuma sincronização:** no qual os nós acessam o canal sem qualquer sincronismo;
- **Sincronização local:** no qual os nós formam um grupo para se comunicar usando uma sincronização ao alcance de somente um salto;
- **Sincronização global:** no qual os nós mantêm a sincronização entre todos os participantes da rede e empregam uma estrutura de vagas (*slots*) de transmissão bem definidas;
- **Operação em multicanal:** no qual os nós podem utilizar mais de um canal de comunicação para a mesma rede.

Pode-se ainda utilizar outras classificações, como é dada por [13], que analisa características para aplicações de missão crítica, em que o atraso e/ou a confiabilidade são muito restritos. Neste caso, foi usado S-MAC [10] (discutido posteriormente) como ponto de comparação e distinção entre os demais protocolos. Nesta classificação, diversos protocolos de comunicação foram essencialmente avaliados quanto ao atraso no transporte de dados e em sua confiabilidade de operação. Para isso, segundo o autor, foram consideradas também as necessidades da aplicação, como:

- **Tolerante ao atraso e tolerante à perda de pacotes:** no caso de aplicações que permitem a perda de pacotes e o atraso. Poucas aplicações se enquadram neste caso, mas há, por

exemplo, os casos de coleta de dados em ambientes hostis ou um simples monitoramento de ambientes;

- **Tolerante ao atraso, mas intolerante à perda de pacotes:** como seria o caso de registros de carros que entram e saem de um determinado estacionamento fechado, para indicar o número de vagas ainda restantes, por exemplo;
- **Intolerante ao atraso, mas tolerante à perda de pacotes:** neste caso pode-se perder dados, contanto que o atraso ainda seja pequeno, como poderia ser visto em transmissões de áudio ou vídeo (com baixo qualidade);
- **Intolerante ao atraso e intolerante à perda de pacotes:** com uma grande restrição tanto em atraso quanto em confiabilidade, como seria o caso de um processo altamente delicado e rápido, como em refinarias de óleo, onde cada minuto parado pode gerar grandes prejuízos.

No caso destas classificações, as aplicações desejadas de automação predial e rural, nos casos específicos deste trabalho, podem ser consideradas como tolerantes ao atraso, mas intolerantes à perda de pacotes, tendo em vista as características já mencionadas anteriormente.

Segundo [13], pode-se ainda classificar cada protocolo individualmente segundo outros aspectos, como quanto a:

- **Importância dos atrasos:** redução do atraso de nó para nó, garantir o atraso máximo de nó para nó, redução do atraso de fim a fim (fonte até destino) e garantir o atraso máximo de fim a fim;
- **Importância da confiabilidade:** aumentar a confiabilidade de nó para nó, garantir a confiabilidade máxima de nó para nó, aumentar a confiabilidade de fim a fim e garantir a confiabilidade máxima de fim a fim.

Considerando que o aperfeiçoamento de uma proposta que utilize transmissões em somente um canal e a tecnologia mais simples (somente o sensoriamento da portadora) possa ser estendido às tecnologias mais atuais (como múltiplas antenas, rádios com *wake-up*, etc), trabalhar em protocolos com esse nível de simplicidade tecnológica pode, na verdade, ser um bom avanço a melhores protocolos de comunicação específicos para determinadas aplicações. Assim, os protocolos utilizados para analisar e comparar com a proposta deste trabalho foram selecionados não só por relevância na literatura, mas também pela utilização de tecnologias mais simples, ou seja, somente sensoriamento da portadora, utilização de ciclos de trabalho com tempos fixos e principalmente se o protocolo foi realmente instalado em um módulo de rádio real. O caso do protocolo TRAMA [14] (energeticamente eficiente e livre de colisões de dados úteis) exemplifica um protocolo de maior complexidade e custo computacional, o que desproporciona a comparação adequada com o proposto protocolo, segundo a aplicação de automação desejada.

Contudo, para avaliação dos protocolos é necessária uma estrutura que se assemelhe à aplicação desejada e, por isso, na literatura foram propostas topologias de operação e avaliação dos métodos

de comunicação capazes de simular tais tarefas. Entre estas, a topologia em estrela que pode ser considerada adequada para simular a automação predial ou rural e a clássica topologia linear que é interessante para fins de comparação com artigos conhecidos na literatura. Dessa forma, nas subseções seguintes serão discutidos trabalhos considerados importantes para a revisão sobre este assunto, mesmo que alguns destes possuam características que desfavoreçam uma comparação ideal com a atual proposta, como uso de tecnologias mais novas, desenvolvimento com fusão de camadas (do inglês *crosslayer design*) e ciclos de ativação variáveis.

2.1 Protocolos de comunicação

2.1.1 Sensor MAC

Entre os protocolos de comunicação atualmente disponíveis que se encaixam relativamente bem aos requisitos desejados, o Sensor-MAC [10] ou também conhecido como S-MAC é uma abordagem consolidada em termos de economia de energia que também foi implementado em uma plataforma real, por meio do sistema TinyOS [15] e módulos *Rene Motes* [16]. Este protocolo trabalha somente com sensoriamento do canal para detecção de congestionamento, bem como características que contribuem para o racionamento, como:

- Uso de ciclo ativo e ciclo inativo (ciclo desacordado, correspondente ao *sleep mode*), correspondente ao ciclo de trabalho (do inglês *duty cycle*) [17] que permite ao sistema realizar períodos de maior economia de energia. Porém esta característica gera um aumento do atraso no envio de mensagens (aumento da latência), e conseqüentemente deve-se configurar o protocolo para melhor atender às aplicações, pois caso o tempo de dormência seja muito grande, pode haver perda por acúmulo de pacotes na memória. Além disso, durante os períodos ativos, todos os componentes da rede ficam ligados e ouvindo/disputando o canal, enquanto durante os períodos inativos do restante da rede, somente dois nós irão se comunicar (os que ganharam na disputa durante o período ativo).
- Pequenos pacotes para controle de tráfego, semelhante ao protocolo de sensoriamento da portadora e múltiplo acesso com prevenção de colisão (CSMA/CA - do inglês *Carrier Sense Multiple Access with Collision Avoidance*) [18] que permite, em caso de uma colisão, baixa perda de energia, bem como reduz o problema de terminais expostos e terminais escondidos. No caso do S-MAC são utilizados, além dos pacotes de controle RTS, CTS e ACK, pacotes de sincronização transmitidos em vagas específicas com contenção para envio destes, ou seja, também podem haver colisões nos pacotes para sincronização.

Uma visão geral do modo de operação deste protocolo pode ser visto na Figura 2.1.

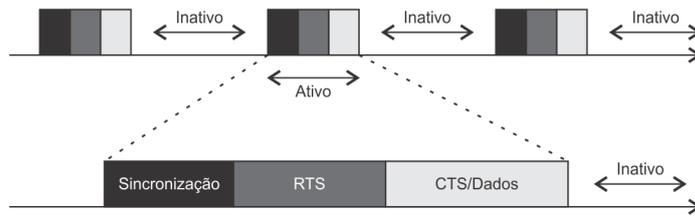


Figura 2.1: Diagrama temporal de funcionamento do S-MAC.

De acordo com [10], o protocolo S-MAC foi desenvolvido para atender à necessidade de economia de energia ao custo do aumento da latência. Contudo este protocolo ainda é baseado na contenção de vagas (*slots*) para transmissão, gerando um certo desperdício.

Outra característica muito importante neste sistema de comunicação é a existência de grupos de ciclos de operação como modo de conexão, ou seja, os nós da rede não se conectam de forma direta a um nó coordenador, na verdade eles escutam o canal e procuram por sinais de rede de algum grupo existente ao redor. Caso haja rede, este pode ouvir por sinais de sincronização que são enviados pelos nós que já estão sincronizados nesta rede existente. Desta forma, são formados grupos de nós comunicantes que operam no mesmo ciclo. Caso um módulo queira enviar uma mensagem para outro de outro grupo, este deve adquirir a agenda de ativação do nó adjacente, por meio dos pacotes de sincronização, e então trabalhar em dois ciclos de ativação diferentes [10].

2.1.2 Sensor MAC com escuta adaptativa

Para resolver o problema de latência no S-MAC, foi desenvolvida também a versão Sensor-MAC com escuta adaptável (S-MAC-AL - do inglês S-MAC *with adaptive listening*) [19]. Nesta nova versão foi também substituída a plataforma de teste, a qual se encontra desenvolvida nos módulos MICA [20] e MICA2 [21]. Entretanto, esta versão não corrige totalmente o problema com a possível alta latência, pois este faz com que os nós da rede consigam transmitir somente um mensagem a mais por período do que a versão anterior S-MAC. Desta forma este protocolo utiliza mais tempo do período de dormência para melhorar a latência do sistema, porém o terceiro salto possivelmente não estará acordado para dar continuidade à transmissão, pois este não estava ao alcance físico do nó que transmitiu no primeiro salto. Portanto o diagrama temporal desta proposta fica como na Figura 2.2.

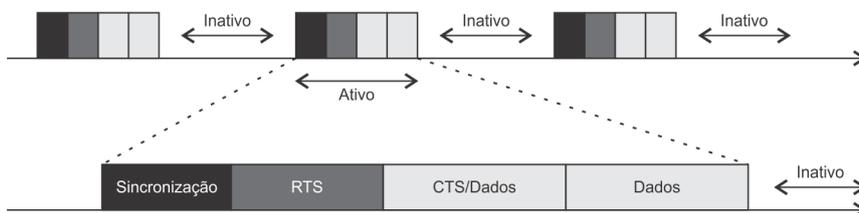


Figura 2.2: Diagrama temporal de funcionamento do S-MAC-AL.

A estratégia do S-MAC impede a colisão de pacotes de dados por meio dos pacotes de controle, porém esse mecanismo pode não ser apropriado para sistemas de automação onde se tem a carga

útil da mensagem de apenas alguns bytes (menor que 100, por exemplo), porque o período de contenção pode ser muito maior do que o necessário para enviar a carga; assim, se gasta mais energia para controlar o canal do que para se transmitir dados.

2.1.3 Múltiplo Acesso por Divisão do Tempo (TDMA)

Considerando o fato de que para as aplicações desejadas, mobilidade e a dinâmica de conectividade não são parâmetros de grande relevância, ou seja, não ocorrem frequentemente nestes casos, o protocolo de múltiplo acesso por divisão do tempo (TDMA - do inglês *Time Division Multiple Access*) [8] permite a utilização do canal de forma organizada e sem colisão. Essa característica é muito interessante para os requisitos listados anteriormente. O funcionamento deste protocolo pode ser visto na Figura 2.3.

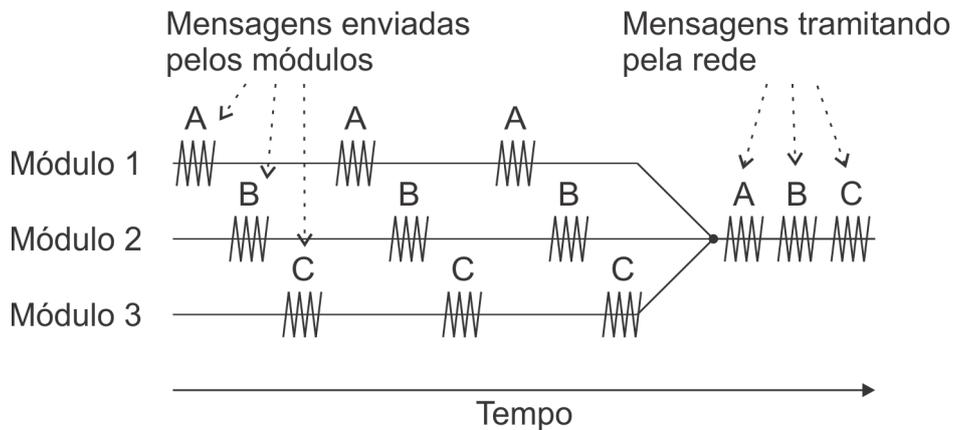


Figura 2.3: Diagrama temporal de funcionamento do TDMA.

Por outro lado, para o funcionamento correto do TDMA, faz-se necessário que os nós da rede saibam quando eles devem entrar em ação para se comunicar, bem como que haja um ou mais destes realizando os sincronismos entre as vagas (em inglês *slots*) de comunicação. Este custo de utilização se torna sua principal desvantagem, pois realizar sincronismo é também uma forma de gasto de energia para o controle da rede.

Outro problema comumente encontrado nesta proposta é em relação à determinação do número de nós conectados, pois uma vez que a rede é criada para determinada quantidade de nós, o número de vagas disponíveis fica fixo. Assim, ou cada nó possui sua vaga, ou eles irão se revezar entre si para utilizar um mesmo *slot*.

A situação pode ainda ser pior quando a rede não está saturada de nós transmitindo, dado que isto possui como consequência um perda de oportunidade de vagas ociosas, gerando com isso uma redução da taxa de transferência e utilização de canal, podendo ainda ser considerado um aumento na latência média, porque são desperdiçadas vagas que poderiam estar sendo utilizadas.

2.1.4 Sensoriamento da portadora e múltiplo acesso com prevenção de colisão (CSMA/CA)

O CSMA/CA (do inglês *Carrier Sense Multiple Access with Collision Avoidance*) [18] é um consolidado protocolo em nível de comunicação sem fio, utilizado em diversas aplicações, bem como muito utilizado em rede de computadores pessoais e celulares. Este protocolo permite a rápida conexão entre os dispositivos e ainda reduz o problema de terminais escondidos e terminais expostos com a utilização de pacotes RTS/CTS seguidos do envio de dados e ACK. Entretanto, este exige um certo nível de consumo de energia, dado que os nós da rede devem estar acordados e aptos a escutar uma requisição de transmissão, bem como também receber e interpretar pelo menos o primeiro passo da negociação necessário na comunicação entre dois módulos. Em alguns casos, deve-se interpretar até mais de um pacote de transmissão alheia, para que se saiba quando o canal estará disponível para utilização. Este funcionamento pode ser visto na Figura 2.4.

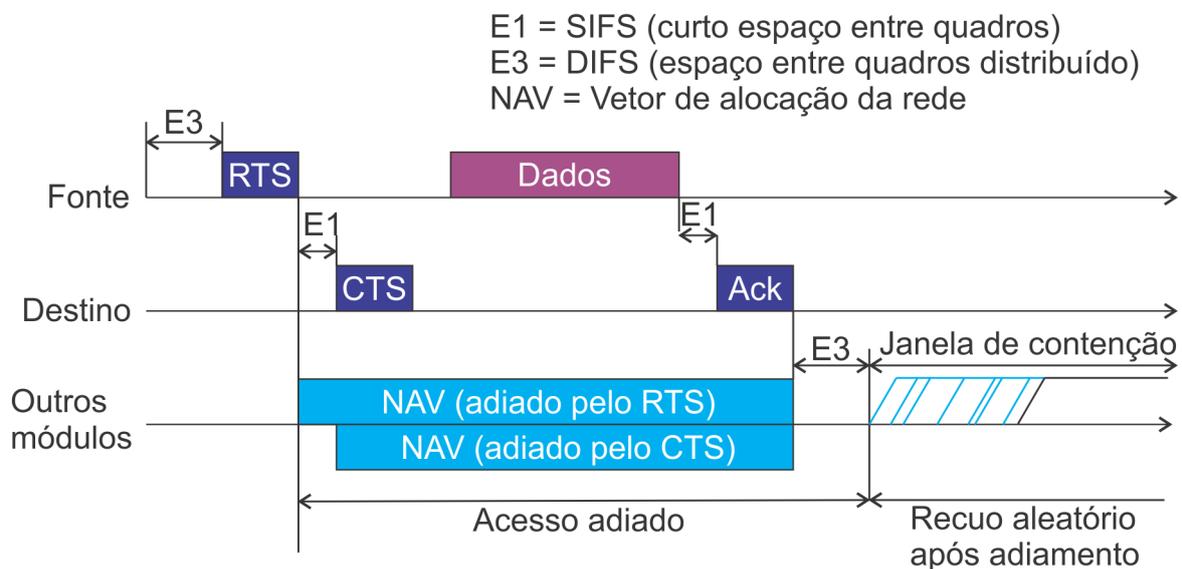


Figura 2.4: Diagrama temporal de funcionamento do CSMA/CA.

Assim, os gastos com colisões entre pacotes de controle (ainda que pequenos) e a necessidade de receber e interpretar dados alheios faz deste protocolo uma solução não muito viável para rede de sensores alimentados por baterias.

2.1.5 Protocolo IEEE 802.15.4

Este padrão de comunicação [11] desenvolvido pelo Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE) permite configuração da camada MAC de forma a melhor atender à aplicação, por meio da determinação de diversos parâmetros, tendo como exemplos de alguns deles:

- Determinar o tempo máximo ativo: por meio do número de vagas para comunicação;
- Determinar o tempo entre sinais de início de período (em inglês *beacon*) ou mesmo se esse

deve existir (é possível o uso deste protocolo no modo mais simples, ou seja, basicamente CSMA);

- Utilização de funções específicas em comunicação, ou seja, habilitar a troca do número de identidade do módulo, permitir ou não notificação de *beacons* à camada superior entre outros;
- Utilização de transmissão de mensagens indiretas ou diretas;
- Utilização de número de identidade com 16 ou 64 bits.

O protocolo IEEE 802.15.4 é o padrão para redes de sensores, capaz de atender variadas aplicações. Dentre as diversas operações realizadas por este protocolo, a mais marcante é a realização dos *superframes*, onde há *slots* disponíveis para cada nó tentar realizar uma comunicação, seja por um processo de contenção (vagas da região CAP, do inglês *Contention Access Period*) ou por *slots* reservados (vagas da região CFP, do inglês *Contention-Free Period*). Os *slots* reservados (região CFP) são divididos entre grupos de GTS (do inglês *Guaranteed Time Slots*), nos quais os nós folhas se comunicam com o coordenador. Cada GTS pode possuir diferentes números de *slots*, mas para cada GTS todos os *slots* possuem sempre o mesmo sentido de transmissão (folha para coordenador ou coordenador para folha), sendo que cada região CFP pode possuir no máximo 7 diferentes GTS [11].

Durante o *superframe* (diagrama da Figura 2.5) ocorrem as mensagens de sincronização, onde também são informados parâmetros da rede, vagas de transmissão com contenção e vagas de transmissão garantidas. O fato de ainda existirem contenções neste protocolo (CAP de inglês *contention access period*), permite a existência de colisões, ou seja, perda de energia.

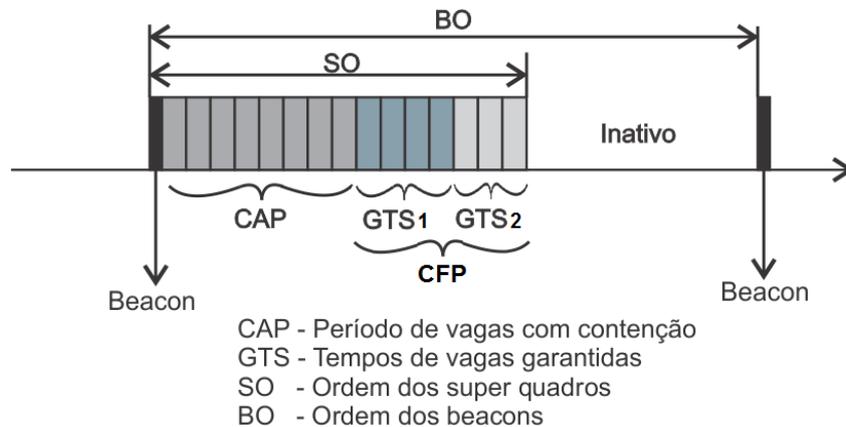


Figura 2.5: Diagrama temporal do *superframe* realizado no protocolo IEEE 802.15.4.

No IEEE 802.15.4 utiliza-se a definição de ordem de *beacons* (BO) e ordem de *superframe* (SO) para determinar o tempo total do período entre *beacons* e o tempo máximo ativo da rede, os quais são dados pelas equações

$$Duração_{Superframe} = aBaseSuperFrameDuration \cdot 2^{SO} \quad (2.1)$$

e

$$\text{Intervalo}_{\text{Beacon}} = a\text{BaseSuperFrameDuration} \cdot 2^{BO}, \quad (2.2)$$

onde $a\text{BaseSuperFrameDuration}$ é um parâmetro do protocolo que, na atual implementação, possui valor de 15,36 ms (960 símbolos, sendo 16 *slots* vezes 60 símbolos por *slot*, assumindo uma taxa de transmissão de dados de 250 kbps, na frequência de 2.4 GHz).

Neste protocolo se utiliza o mesmo e principal meio de interação do protocolo S-MAC, ou seja, somente o sensoriamento do canal. É importante ressaltar que este sensoriamento da portadora (CCA - do inglês *Clear Channel Assessment*) pode ser feito de três diferentes maneiras, como:

- **Leitura da energia do canal acima do limiar:** Realizado por um processo de detecção do nível de energia em determinado canal (ED - do inglês *energy detection*), caso o resultado seja maior que um determinado limiar, entende-se que o canal esteja ocupado por uma transmissão alheia;
- **Sensoriamento simples da portadora (CS - do inglês *Carrier sense*):** Processo que avalia o canal de comunicação por meio de análise das mensagens circulando pelo meio, ou seja, será dado um resultado positivo caso seja detectado alguma mensagem pertencente ao padrão do protocolo, independentemente do resultado da verificação de ED.
- **Combinação das duas anteriores:** Neste caso, é gerado um resultado positivo se e somente se houver resultado positivo tanto na leitura de ED, quanto na leitura de CS.

Além disso, o processo de conexão deste protocolo é mais estrito à relação de coordenador, roteadores e folhas (ponta da rede). Assim, geralmente não se formam grupos independentes na mesma rede, como no caso do S-MAC, mas sim uma grande rede composta desses três tipos.

2.1.6 T-MAC

Na proposta do *Timeout Medium Access Control* (T-MAC) [22] foi feita uma melhoria nos parâmetros de latência e consumo de energia gastos em períodos ociosos, ou seja, reduz-se períodos de esperar por pacotes quando a rede não está congestionada o suficiente para utilizar todo o período ativo disponível. Outro ponto importante nesta proposta é o fato de também ter sido implementado em módulo real EYES hardware [23].

Desta forma, o T-MAC [22] traz uma melhoria para sistemas que podem necessitar de uma certa economia de energia, porém tem como prioridade uma vazão relativamente razoável e uma latência menor que a observada no S-MAC. Isso é feito por meio de um algoritmo para aumentar ou reduzir o tempo ativo total em que os módulos irão operar, variando de acordo com as chegadas de requisições de transmissão. Segundo [22], o T-MAC atinge, com uma mesma vazão, menores consumos, comparado ao S-MAC, em sistemas que possuem taxa de chegada muito variantes e no mínimo desempenho semelhante em sistemas com taxas constantes. Um exemplo de funcionamento deste protocolo pode ser visto na Figura 2.6.

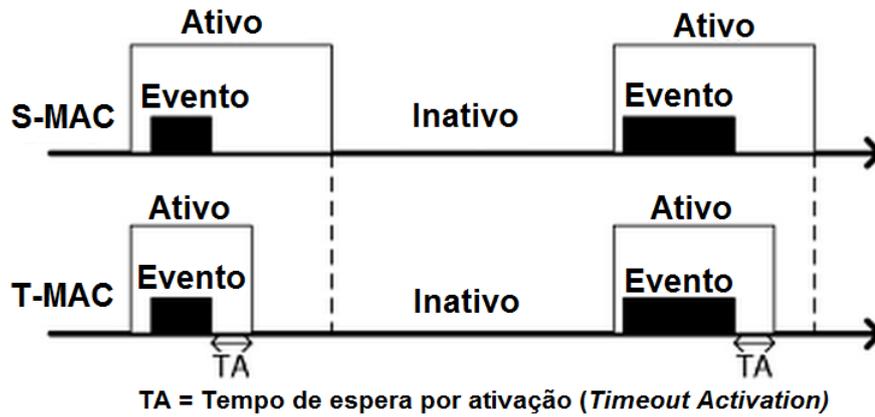


Figura 2.6: Diagrama temporal de funcionamento do T-MAC em comparação com S-MAC.

Como pode ser visto no diagrama comparativo, o T-MAC possui um tempo ativo variável de acordo com a duração dos eventos do grupo, adicionado de um tempo de escuta ociosa de segurança (TA). Isso permite que o protocolo opere em diferentes pontos de operação da rede, com menores perdas de energia em períodos ociosos.

Contudo, neste protocolo ainda existem perdas por colisões e a utilização de pacotes de controle (RTS, CTS) e pacotes de sincronização. Por um lado, isso favorece evitar problemas com terminais escondidos e terminais expostos, mas por outro lado também favorece ao aumento de excesso de cabeçalho (em inglês *overhead*) e, conseqüentemente, maior gasto com energia neste aspecto.

A forma de conexão nesta sugestão é semelhante ao S-MAC, não havendo, portanto, uma relação direta de coordenador, mas havendo nós realizando sincronização semelhantemente ao S-MAC.

2.1.7 WiseMAC

Trazendo uma abordagem diferente das vistas anteriormente em rede de sensores, o WiseMAC [24] considera seu foco de aplicação em ambientes que possuem pontos de acesso à rede e estão fisicamente conectados à fonte de energia, ou seja, tomadas de eletricidades, etc. Assim, neste trabalho, para a economia de energia é considerado que o maior gasto venha de períodos ociosos escutando o canal (em inglês *idle listening*) e, portanto, tem-se como principal característica deste trabalho a redução destes períodos.

De fato, em ambiente com poucas transmissões, ou seja, baixa geração de mensagens, a escuta do canal ocioso se torna um grande problema, mas nem sempre é o caso de demais tipos de aplicações. No entanto, esta proposta obtém bons resultados para sistemas específicos e foi base para outros protocolos utilizados em rede de sensores alimentados por bateria que serão discutidos posteriormente. Dado isto, torna-se importante analisar e estudar também seus aspectos de funcionamento, como pode ser visto na Figura 2.7.

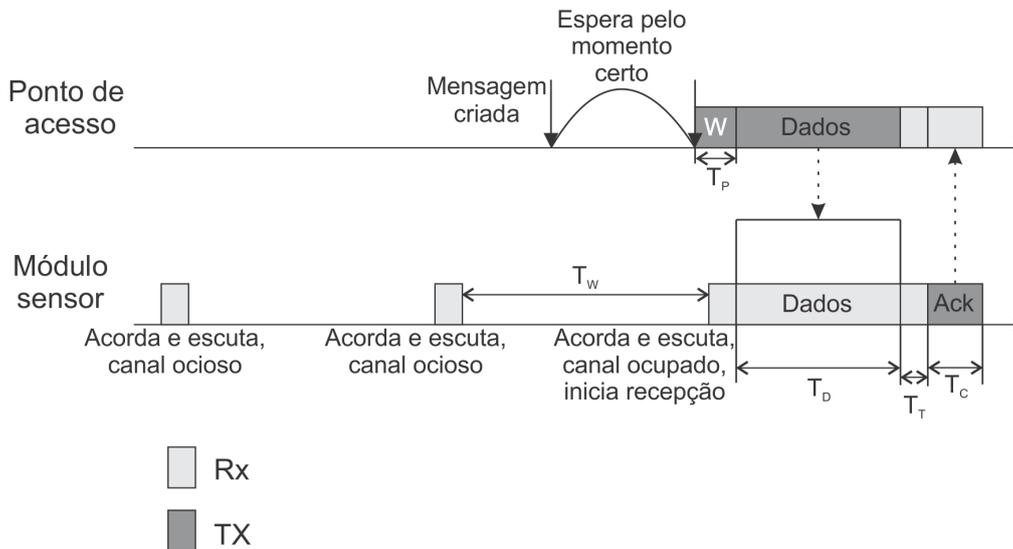


Figura 2.7: Diagrama temporal de funcionamento do WiseMAC.

Na Figura 2.7, pode-se observar o uso de preâmbulo (pacote “W”) para a transmissão de dados, ou seja, cada nó da rede realiza verificações temporárias do canal (períodos T_W) a espera de uma mensagem inicial (preâmbulo). Na medida em que um nó transmissor gerar mensagens, esta irá esperar um tempo ideal para iniciar o preâmbulo. Assim que o tempo “W” acabar, inicia-se a transmissão de dados (duração T_D), considerando que neste processo foi garantido que o receptor já tenha ouvido o pacote “W”. Após a transmissão dos pacotes desejados e um tempo de troca de função do transmissor (T_T), realiza-se o envio de um pacote de reconhecimento (ACK que tem duração fixa T_C).

Segundo o autor, a utilização de preâmbulo reduz a vazão máxima do sistema em casos de alta taxa de geração de mensagens, ainda que neste protocolo de comunicação ocorra o fato de que todos os nós ao redor do transmissor irão ter gastos desnecessários com a recepção dos dados enviados por ele a um só desses nós. A principal contribuição do WiseMAC foi a tentativa de redução de períodos T_P de escuta “W” com base na taxa de amostragem do canal realizadas pelos demais nós, ou seja, a realização de um certo tipo de sincronismo entre estes para uma melhor economia.

2.1.8 B-MAC

Também desenvolvido em Berkeley (como [10] e [19]), nesta proposta [25], o foco do protocolo está em aplicações de monitoramento e de tráfegos variáveis. Neste âmbito, este sistema não utiliza mais sincronismos, mas sim um preâmbulo grande o suficiente para iniciar uma transmissão. Desta forma, ao se iniciar o envio de um pacote, inicia-se também o envio de um grande pacote de controle, somente para que o receptor tenha tempo suficiente de recebê-lo e responder com uma confirmação de que está pronto. Este preâmbulo só ocorre quando há mensagem para se transmitir, nos demais tempos o sistema permanece dormindo ou realizando pequenas verificações de recepção.

Considerando esta diferente abordagem, onde não se encontram relações diretas em coordenador

e folha, bem como não ocorrem períodos de ativação em grupo, pode-se perceber que a economia de energia está relacionada principalmente à redução de escutas ociosas do canal, se obtendo um bom resultado em situações de baixa contenção do canal. Desta forma, para aplicações de automação predial e rural, não se enquadra como um bom protocolo de comunicação, dado os requisitos citados anteriormente. Isso pois, estas podem possuir uma alta taxa de geração de mensagens, porém de baixo tamanho e, considerando que o preâmbulo seria muito maior que o próprio pacote útil, haveria neste caso um grande *overhead* e consumo de energia desnecessariamente.

No entanto, considerando as aplicações alvo do B-MAC, neste foi desenvolvido um interessante método de escuta de canal (CCA) chamado de LPL (do inglês *Low Power Listening*), onde se economiza bastante o gasto com períodos ociosos de escuta sem transmissão de dados. Existe ainda a possibilidade de utilização de pacotes de controle RTS/CTS semelhantes ao S-MAC, de forma a se evitar os problemas de terminais escondidos e expostos, mas esta configuração reduz em muito a vazão e outros parâmetros.

Contudo, um dos grandes benefícios do B-MAC, visando aplicações de monitoramento, encontra-se em sua simplicidade e baixo custo computacional, o qual segundo [25], o B-MAC implementado nos módulos MICA2 [21] com suas diversas configurações é apresentado na Tabela 2.1.

Tabela 2.1: Resultados do tamanho da implementação, em bytes, dos protocolos B-MAC e S-MAC em motes reais.

Protocolo	ROM (Bytes)	RAM (Bytes)
B-MAC	3046	166
B-MAC com ACK	3340	168
B-MAC com LPL	4092	170
B-MAC com LPL e ACK	4386	172
B-MAC com LPL,ACK e RTS-CTS	4616	277
S-MAC	6274	516

2.1.9 X-MAC

Seguindo a mesma linha de raciocínio presente no WiseMAC [24] e no B-MAC [25], a abordagem do X-MAC [26] traz uma melhoria significativa em termos de economia de energia com a escuta ociosa, pois este propôs que os preâmbulos fossem na verdade pequenas transmissões periódicas capazes de serem captadas pelo nós receptor.

Neste caso, ganha-se em economia não só porque os nós estão ativos por menos tempo, mas também pelo fato de que no envio dos pequenos preâmbulos estão presentes os números de identificação (ID). Dessa forma, os nós presentes ao redor do transmissor irão não só detectar que o canal possui um preâmbulo sendo enviado, como também identificar que a mensagem não será para ele, assim indo dormir logo em seguida, ao contrário do que ocorre no B-MAC, onde todos os nós esperam pelo dado, para então identificar que não pertence a ele e ir dormir. Assim, a diferença de funcionamento entre estes protocolos é apresentada na Figura 2.8. Sendo assim, um grande avanço desta proposta foi a melhoria da economia em redes mais densas.

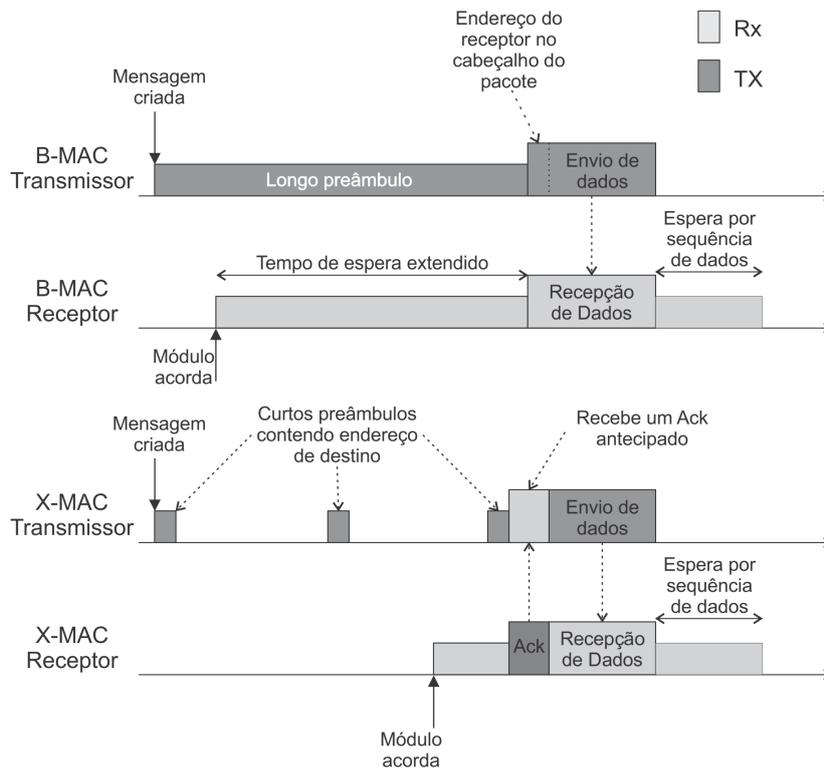


Figura 2.8: Diagrama temporal comparativo entre o funcionamento do B-MAC e X-MAC.

2.1.10 AEDP e outros protocolos com base em LPL

Diversas outras variantes de protocolos que utilizam LPL e préambulos para a inicialização de transmissões foram propostas na literatura, como o BoX-MAC-2 [27] que tem como base o X-MAC [26]. Este substituiu o campo de endereço presente no préambulo (X-MAC), de forma a enviar pacotes de dados úteis já no lugar deste como forma de préambulo.

Existem também as propostas que consideram o início da comunicação dado pelo receptor (em inglês *receiver-initiated MAC protocols*), como o A-MAC [28]. Neste foi observado que o problema de desperdício com o falso início de transmissão (em inglês *false wakeup*) no canal pode ser reduzido se o módulo iniciador da comunicação for o receptor, pois este somente irá enviar préambulos quando estiver pronto para receber.

Dessa forma, o BoX-MAC-2 e o A-MAC serviram de base para o AEDP (do inglês *Adaptive Energy Detection Protocol*) [29], o qual possui comunicação iniciada pelo transmissor (em inglês *sender-initiated*) e propõe uma opção de melhor eficiência no caso de redes menos congestionadas. Isso pois o AEDP evita o excesso de cabeçalho dos pacotes de controle utilizados nos demais protocolos *receiver-initiated*.

A principal contribuição do protocolo AEDP [29] foi no estudo do impacto da taxa de *false wakeups* dados os limiares de potência utilizados durante o processo de CCA (escuta e verificação do canal). Dessa forma este protocolo propõe um sistema dinâmico capaz de melhor redefinir os limiares de CCA que serão utilizados pelos módulos, reduzindo assim o consumo pela menor taxa

de falsas recepções. É importante citar que o AEDP foi implementado em módulos reais utilizando o compacto sistema TinyOS [30] e módulos TelosB [31].

2.1.11 DMAC

Em alguns trabalhos, a proposta de economia de energia visando determinada aplicação extrapolava o nível da camada MAC, como é o caso do protocolo DMAC [32]. Neste foi realizado um desenvolvimento em mais de uma camada de comunicação (em inglês *cross-layer design*), no caso a camada MAC e a camada de roteamento. Este desenvolvimento em múltiplas camadas proposto pelo DMAC [32] foi um dos motivadores de trabalhos futuros apresentados no Capítulo 6, visto que a adaptação de protocolos de roteamento com foco em eficiência energética [33] integrado à camada MAC pode fornecer um maior desempenho no sistema, se comparado a módulos que operam com camadas e protocolos independentemente desenvolvidos.

A principal aplicação do DMAC é em sistemas que necessitam de um baixo atraso nas mensagens, bem como uma boa economia de energia. Entretanto, devido ao modo como foi proposto (estilo conexão em árvore), este protocolo é específico para sistemas que trabalhem com um só destino (sumidouro de dados, como uma central receptora) e demais ramos sensores e transmissores. Isso pois, cada nó da rede está diretamente associado (devido ao projeto de roteamento determinado para este protocolo) a outro nó em um nível superior, sendo que este irá retransmitir suas mensagens até que esta chegue ao destino único e final. A Figura 2.9 apresenta a forma de funcionamento deste protocolo.

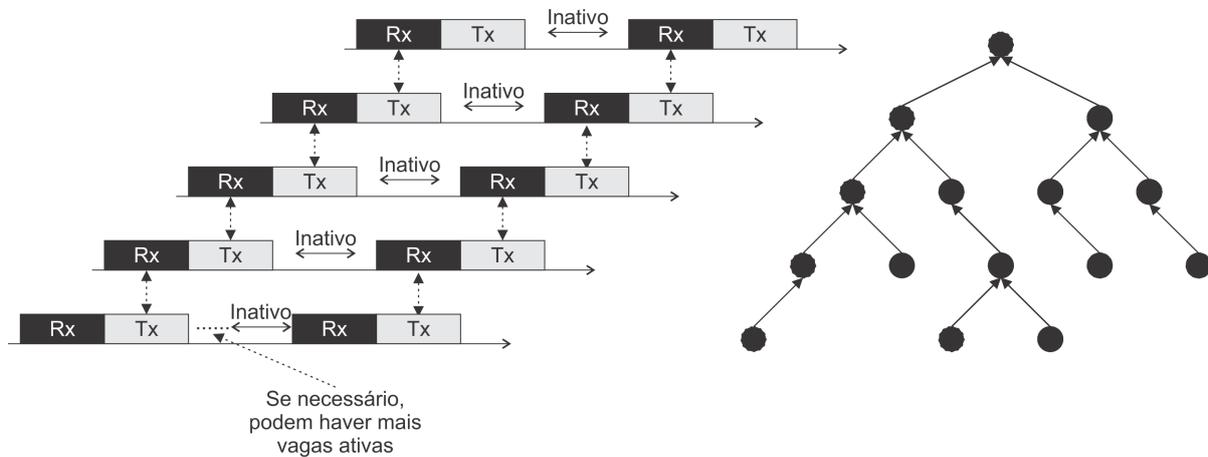


Figura 2.9: Diagrama temporal e estrutura de roteamento do DMAC.

No DMAC também é feita uma adaptação dos períodos de ativação, de forma que na medida em que os pacotes comecem a chegar mais rápido, os nós comecem a trabalhar mais vezes, mantendo a QoS e garantindo melhor eficiência. Além disso, como pode ser visto na Figura 2.9, os nós em níveis mais baixos iniciam seus períodos antes dos nós de níveis superiores, de forma que quando o nível inferior acabar de receber o pacote, este e o nível superior já estejam aptos para comunicarem, fazendo assim uma rota “direta” dos dados até o sumidouro.

Contudo, apesar da eficiência deste protocolo, aplicações que necessitem de transmissão de dados entre os nós participantes da rede não irão conseguir operar com o DMAC. Sendo assim, este protocolo não é exatamente um concorrente do S-MAC, no qual cada nó tem a liberdade de retransmitir ou enviar mensagens para quaisquer outros nós da rede. Vale citar ainda que no DMAC ainda ocorrem períodos de contenção, ou seja, é possível presenciar colisões, as quais prejudicam a economia de energia.

2.1.12 TDMA-Wakeup

Outras propostas na área de rede de sensores consideram a utilização de tecnologias adicionais para a comunicação. Podendo variar em diversas formas, como:

- **Wake-up hardware:** permite acordar os nós, via interface sem fio, sem interferir no canal utilizado;
- **Sincronização via GPS:** permite sincronizar os relógios dos módulos por meio de outra tecnologia chamada sistema de posicionamento global (GPS - do inglês *global positioning system*);
- **Antenas direcionais:** permitem o envio de dados para uma só direção, evitando interferências radiais. Entretanto, nesta opção de comunicação é necessário o conhecimento prévio da posição de cada outro módulo, seja por tabelas pré definidas, seja por outras tecnologias (GPS, por exemplo).

No caso do TDMA-W (TDMA-Wakeup) [34], foi utilizado o conceito, já discutido, sobre TDMA associado da consideração de uma sincronização de relógio via GPS. Assim, o protocolo foi capaz de obter bons resultados, mas não chegou a ser implementado em módulos reais para testes mais apurados.

2.1.13 ID-MAC

Neste protocolo é proposta uma estrutura de funcionamento capaz de organizar uma rede de nós sem fio sincronizada e sem necessidade de um nó coordenador da rede [35]. Assim, pode-se ter os benefícios de uma rede com sincronização e slots (pouca ou nenhuma colisão), bem como melhor eficiência energética, visto que não se tem tantos gastos com pacotes de controle como nos demais protocolos. É importante citar que no ID-MAC existe a situação em que dois ou mais nós podem colidir seus pacotes, a qual ocorre quando a rede não possui um único caminho de dados, assim um nó pode receber pacotes de mais de um transmissor.

Esse mecanismo é proposto por meio de funções pré-programadas associadas ao número de identidade único de cada módulo. Assim, dado seu endereço e dos demais módulos vizinhos a ele, o módulo saberá quando ele deve escutar o canal para receber transmissões e enviar pacotes entre ele e seus vizinhos. O ID-MAC [35] foi desenvolvido e divulgado somente na plataforma de simulação NS-2 [36].

Entretanto o funcionamento do ID-MAC está baseado em um pré-conhecimento dos nós vizinhos, o qual também necessita de pacotes de controle para este reconhecimento. É considerado também que todos os nós estejam sincronizados, para que operem em turnos de duração fixa (períodos). Assim, durante um turno ocorre um instante de *broadcast* e outros instantes de recepção e transmissão de cada módulos da rede.

A estrutura de sincronização, segundo o autor [35], pode ser dada por um processo de referência do *broadcast*, porém, visto que o protocolo não foi implementado em módulos reais, não é possível afirmar a eficiência deste processo de sincronismo, bem como garantir então a exatidão da operações dos turnos, visto o desvio de relógio (do inglês *clock drift*) sofrido pelos módulos.

O conhecimento prévio dos módulos vizinhos deste protocolo faz com que se exija uma complexa troca de informações entre os nós, bem como o armazenamento de todos os endereços em memória. Considerando um sistema mais denso de módulos, este processo pode ser caótico e de alto custo computacional e energético. Outro gasto de energia no ID-MAC pode ser dado pelos problemas de terminal exposto e terminal escondido, os quais não foram estudados neste trabalho. Ainda, considerando uma estrutura em malha (*mesh*), devido ao fato de não possuir coordenador, no ID-MAC todos os nós tem que acordar nas vagas de transmissão de cada outro vizinho, pois não há como saber se este deseja ou não enviar pacotes para ele. Logo ocorre também um grande número de momentos ativos durante um turno, de forma a receber uma possível transmissão.

Por fim, segundo o autor [35], o ID-MAC apresentou somente até 15% a menos consumo de energia do que o S-MAC[10]. Dessa forma, uma aplicação que permita determinar a posição de cada módulo, bem como utilizar parâmetros fixos pode ser um bom foco deste protocolo, visto que este pode reduzir o consumo de forma mais eficiente nestes casos.

2.1.14 AS-MAC

O desenvolvimento deste protocolo foi baseado no padrão IEEE 802.15.4, bem como o módulo utilizado para os experimentos possuiu especificações próprias para este padrão. Assim, a proposta do AS-MAC [37] foi bem próxima ou praticamente uma melhoria do protocolo IEEE 802.15.4. Isso pois, um dos grandes benefícios do AS-MAC foi o desenvolvimento de melhores técnicas de percepção da qualidade do canal, permitindo adaptar outros parâmetros para operarem de forma ótima.

Assim, o envio de pacotes foi alterado de forma que enquanto no padrão IEEE 802.15.4 utiliza-se uma troca de pacotes de controle (RTS e CTS) para negociar o início da transmissão de um único pacote, no AS-MAC tem-se a primeira mensagem utilizada para iniciar a comunicação já contendo carga útil e cabeçalho estendido, ou seja, exercendo a função do RTS. Após a recepção da confirmação, com dados suficientes para inferir a qualidade do canal (ACK acrescido de informações da recepção), é feita uma aglomeração dos pacotes restantes na fila, de forma a enviá-los sequencialmente e de uma só vez.

Dessa forma, a contribuição deste protocolo está na decisão da quantidade de pacotes enviados neste processo de aglomeração, sendo este variável de acordo com a qualidade do canal, bem como

o método de codificação da transmissão (faz-se necessário o uso de módulos capazes de alterar sua modulação e, assim, a quantidade de bits por símbolo), o qual permite usufruir de melhores vazões e economia de energia dependendo da relação sinal ruído e qualidade da conexão no canal.

Contudo, este protocolo emprega técnicas de aprimoramento com funcionalidades não desejadas para a comparação com os demais trabalhos, visto que a alteração da modulação utilizada também pode ocasionar em melhores resultados no outros protocolos. Tem-se ainda que a aglomeração de pacotes pode não ser uma técnica desejada em algumas aplicações, pois pode aumentar a latência do sistema, ainda que este processo reduza a sobrecarga de cabeçalho da rede.

2.2 Topologias utilizadas

Para realizar a análise dos protocolos desenvolvidos é necessário que se determine uma configuração de comunicação entre os nós, chamada de topologia de comunicação. Na literatura ([10, 19, 25, 32, 37] e outros), para melhor avaliar um ambiente de WSN é necessário se adotar uma topologia que se assemelhe à desejada aplicação. Desta forma, foram propostas diversas topologias ao longo da literatura, cada uma possuindo características específicas que permitiriam obter resultados próximos do quais seriam obtidos se a WSN fosse de fato instalado naquela aplicação.

As principais topologias atualmente utilizadas são:

- **Linear:** onde os nós transmitem mensagens em uma sequência determinada, passando-as de um para o outro linearmente. Nesta configuração, pode-se testar os atrasos para as recepções entre múltiplos saltos, bem como a simulação de ambientes mais congestionados quando considerado que todos os nós se escutam (estão dentro do alcance do outro).
- **Malha (em inglês *mesh*):** onde cada nó pode enviar e receber mensagens dos outros nós na rede, sem haver um concentrador de mensagens, ou seja, cada nó transmite para qualquer outro nó que esteja ao seu alcance, sem necessariamente ter que passar sua mensagem por um coordenador ou roteador;
- **Estrela:** onde cada nó está diretamente conectado a um coordenador comum;
- **Árvore de grupos (em inglês *cluster-tree*):** onde os participantes estão divididos em níveis de comunicação, possuindo os nós folha, que geralmente são os sensores, os nós roteadores, que podem atuar como sensores e retransmissores de dados e o nós coordenadores que geralmente só realizam a coleta dos dados dos demais nós;
- **Combinação de estrela e árvore de grupos:** que seria o caso de haver um sistema que possua um ambiente administrado por um nó formando uma árvore de grupos e, após coletar os dados, retransmite para outra central segundo uma topologia estrela.

Outras topologias também foram utilizadas em outros trabalhos para exemplificar situações específicas, como é o caso da topologia linear, proposta no trabalho [10]. Considerando a importância do S-MAC no meio acadêmico, esta topologia foi replicada em alguns outros trabalhos para que

fosse possível se comparar os resultados entre eles. Dessa forma, neste projeto serão utilizadas duas topologias somente, a topologia linear, por motivos de comparação com o trabalho do S-MAC [10], e a topologia estrela, por sua simplicidades e adequação a sistemas de automação.

2.2.1 Linear

A topologia linear na Figura 2.10 foi utilizada no protocolo S-MAC, de acordo com os autores em [10], sendo esta rede composta de alguns poucos nós (no caso de [10], foram utilizados dez módulos) e um destinatário. Nesta topologia, foi considerado que todos os nós escutam cada um dos outros, ou seja, cada nó está dentro do alcance de transmissão do outro, mas somente o primeiro nó (nó origem) gera novas mensagens em intervalos regulares. Assim, durante os períodos ativos de um determinado protocolo, cada participante da rede só tenta transmitir sua mensagem ao seu vizinho adjacente em direção ao destino, até que a mensagem chegue no destinatário, que atua apenas como um receptor. Desta forma, essa rede possibilita simular uma situação de múltiplos saltos de mensagem, aumentando a disputa pelo canal, sem que seja necessário o uso de muitos módulos, os quais estão somente conectados à sua própria bateria.

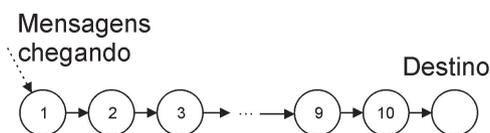


Figura 2.10: Topologia linear.

A finalidade desta topologia não é exatamente simular uma aplicação real, mas um caso específico recorrente em WSN, onde se necessita retransmitir uma determinada mensagem por vários nós. Pode-se, por exemplo, enxergar esse sistema dentro de um caso dentro da topologia de árvore de grupos. Assim, a forma linear de avaliar uma rede é na verdade um procedimento para se averiguar situações de atrasos em múltiplos saltos, sem ter que utilizar para isso uma grande quantidade de módulos operantes, o que torna mais viável para implementação real. Contudo, para também se considerar problemas de contenção, considera-se essa rede da forma que cada módulo esteja ao alcance do outro.

2.2.2 Estrela

Considerando um sistema de automação mais simples, como o exemplo da Figura 1.2(a), a topologia em estrela permite simular melhor e de forma simples o trajeto e geração de mensagens deste sistema. Isso pois, em um ambiente com WSN visando a automação, como seria o caso de sensores de temperatura, atuadores em um ar condicionado e sensores de abertura de portas em um escritório, cada nó da rede irá gerar mensagens independentemente, seja em intervalos constantes ou em momentos aleatórios (chegadas com probabilidade Poisson).

Neste caso, a topologia estrela considerada é também composta de alguns poucos módulos e um nó central (destinatário final), onde todos os participantes da rede estão ao alcance dos demais,

bem como são alimentados por suas baterias independentemente. A Figura 2.11 apresenta esta topologia.

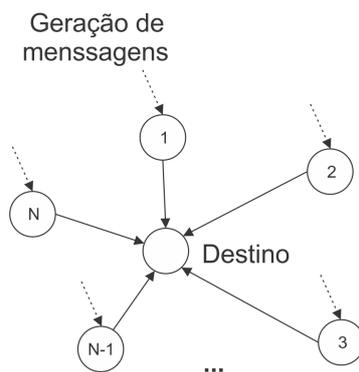


Figura 2.11: Topologia estrela.

No caso da estrutura em estrela é possível observar o sistema “competindo” para enviar mensagens para um só nó, isso possibilita avaliar não só a competição da rede por *slots* de transmissão, mas também a competição pela escuta de um determinado nó. O fato de todos os nós se escutarem também congestiona ainda mais a rede. Assim, pode-se simular um ambiente com problemas complicados de se resolver do ponto de vista sem fio.

Em automação predial o nó destino é geralmente o receptor para o supervisorio (SCADA, do inglês *Supervisory Control and Data Acquisition*), que registra os dados de interesse e dispara tarefas de automação (e.g., liga ar condicionado). Nestes sistemas pode-se ter mensagens transmitidas frequentemente para o monitoramento, bem como mensagens transmitidas algumas poucas vezes para a configuração de pontos de operação, por exemplo.

2.3 Quadro comparativo

Os protocolos vistos anteriormente, sob uma perspectiva geral, podem ser comparados por suas várias características, na forma da Tabela 2.2. Nesta, cada protocolo é marcado na característica que possui entres as diversas observadas. É importante observar que a sincronização pode ser feita de várias formas, mas as principais observadas foram as realizadas por um nó líder (centralizada), por uma troca de mensagens entre os nós do grupo (Distribuída) ou por outros meios e tecnologias, como GPS (*Global Positioning System*). Nesta Tabela 2.2 pode-se observar também a definição de colisões de pacotes englobando todos os tipos de pacotes que possam colidir no protocolo, exceto os decorrentes do processo de registro em grupos (indicado na tabela pelo *), os quais geralmente são feitos uma única vez ou poucas vezes nas aplicações de automação desejadas com baixa ou nenhuma mobilidade dos módulos.

Tabela 2.2: Tabela comparativa entre os protocolos apresentados neste capítulo.

Protocolo	Característica						
	Implementação: - Simulado - Módulo real	Sincronização: - Não realizada - Centralizada - Distribuída - Outros meios	Transmissão: - Preâmbulo: Longo (L) Curto (C) - Slots definidos - Slots aloha - Aloha	Ciclos de trabalho: - Não possui - Fixos - Variáveis	Complexidade: - Simples - Média - Complexo	Usa tecnologia adicional ou cross-layer	Colisão de pacotes (dados ou controle)
DyTEE	Real	Centralizada	Slots definidos	Fixos	Média	Não	Não*
S-MAC	Real	Distribuída	Slots definidos	Fixos	Simples	Não	Sim
S-MAC-AL	Real	Distribuída	Slots definidos	Variáveis	Simples	Não	Sim
TRAMA	Simulado	Distribuída	Slots definidos	Variáveis	Complexo	Não	Sim
TDMA	Real	Centralizada	Slots definidos	Fixos	Simples	Não	Não
CSMA/CA	Real	Não realizada	Aloha	Não possui	Média	Não	Sim
IEEE 802.15.4	Real	Centralizada	Slots definidos	Fixos	Média	Não	Sim
T-MAC	Real	Distribuída	Slots definidos	Variáveis	Média	Não	Sim
WiseMAC	Simulado	Não realizada	Preâmbulo (C)	Fixos	Simples	Não	Sim
B-MAC	Real	Não realizada	Preâmbulo (L)	Não possui	Simples	Não	Sim
X-MAC	Real	Não realizada	Preâmbulo (C)	Fixos	Média	Não	Sim
AEDP	Real	Não realizada	Preâmbulo (C)	Fixos	Média	Não	Sim
Box-MAC	Real	Não realizada	Preâmbulo (C)	Fixos	Média	Cross-layer	Sim
A-MAC	Real	Não realizada	Preâmbulo (C)	Fixos	Média	Sim	Sim
DMAC	Simulado	Distribuída	Slots definidos	Variáveis	Complexo	Cross-layer	Sim
TDMA-W	Simulado	GPS	Slots definidos	Fixos	Média	Sim	Não
ID-MAC	Simulado	Distribuída	Slots definidos	Fixos	Complexo	Não	Não
AS-MAC	Real	Centralizada	Slots definidos	Fixos	Complexo	Sim	Sim

2.4 Conclusão do capítulo

Neste capítulo foram abordados diversos trabalhos na literatura, os quais desenvolvem protocolos com diferentes características e aplicações. Dessa forma, foram analisados em seus aspectos positivos e negativos, os quais também podem ter contribuído para o desenvolvimento do protocolo proposto por este trabalho.

A principal característica observada nestes protocolos foi a distinção entre utilizar ou não um método de sincronização da rede. Isso permite classificar os protocolos entre sincronizados e não sincronizados. Nos que realizam sincronismo, percebe-se também a definição de vagas de transmissão que reduzem ou eliminam as colisões da rede, porém aumentam o consumo com pacotes de controle e sincronização do grupo. Nos trabalhos que optam por sistemas assíncronos, percebe-se o aumento do tempo total em modo de escuta do canal, geralmente ocioso, bem como o uso de preâmbulos, que entram como pacotes de controle e conseqüentemente gasto de energia, juntamente com as possíveis colisões de dados comumente presentes nestes protocolos.

Pode ser visto também que algumas sugestões contam com mecanismos ou tecnologias externas, ou seja, sincronização por outros componentes, métodos de conhecimento prévio da posição dos módulos etc. Estes trabalhos não podem ser considerados como uma justa comparação com os protocolos que utilizam somente procedimentos básicos (sensoriamento do canal e transmissão de pacotes de controle). Assim se faz necessário distingui-los neste aspecto.

De uma forma geral, os testes de protocolos não podem ser feitos sem uma estrutura de comunicação. Desse modo, foram propostos também topologias que permitem simular aplicações que possuem semelhantes formas de operação, sendo a topologia estrela a mais simples e ainda próxima das aplicações de automação predial e rural.

Capítulo 3

Protocolo *Dynamic Timed Energy Efficient* (DyTEE)

Este capítulo apresenta o protocolo desenvolvido neste trabalho, especificando os detalhes entre cada etapa necessária para a comunicação de uma rede de sensores e atuadores.

Visando atender melhor a sistemas de automação predial e rural com as características citadas na Subseção 1.3, o protocolo *Dynamic Timed Energy Efficient* (DyTEE) foi proposto com base nas melhores características observadas nos demais protocolos pesquisados, como o S-MAC [10], TDMA [8] e outros. No caso destas aplicações, a escuta excessiva do canal ou escuta ociosa é também um grande causador de desperdício de energia, mas acompanhado do fato de que um mínimo cabeçalho necessário para as negociações sem fio já podem ocupar uma grande porcentagem do total de bytes enviados no pacote. Considerando um sistema de automação que possua como carga útil de somente 20 bytes, o uso de 7 bytes de cabeçalho já corresponde a aproximadamente 26% do total de bytes enviados. Assim uma forma de economia de energia nesses sistemas encontra-se também na forma de redução de cabeçalhos utilizados, bem como pacotes de controle.

O DyTEE opera de forma cíclica, em termos de períodos de ativação dos nós, e possui uma metodologia TDMA para evitar colisões de dados, garantindo uma boa vazão ao custo de uma maior latência. Segundo os critérios citados por [12], o DyTEE pode ser classificado como do tipo com *sincronização global* empregando a operação de um único canal. Considerando ainda outra classificação explicitada por [13], o DyTEE pode ser classificado também como por *garantir a entrega de nó para nó pelo pior caso de atraso*, ou seja, os dados chegarão ao destino, mesmo que seja ao custo de um alto atraso. O atraso máximo depende dos parâmetros adotados pela rede, bem como da quantidade de nós conectados. Em uma rede com somente 1 nó folha conectado em um coordenador, o atraso máximo é dado pela parâmetro de ciclo de trabalho, ou seja, pelo tempo de período necessário para a rede realizar um ciclo completo (ativo e inativo).

Em toda comunicação sem fio, existe a possibilidade de que se perca o pacote transmitido. No caso do DyTEE, a perda de pacotes é considerada nos casos em que é detectada uma falha na recepção (verificador de erros acusa falha) ou quando o tempo de espera pela confirmação (Ack,

por exemplo) ultrapassa um certo limiar, ou seja, expira o tempo de espera (em inglês *timeout*). Nestes casos, outro parâmetro da rede determina qual atitude deve ser tomada. Dependendo das configurações do protocolo, a mensagem perdida pode ser retransmitida automaticamente em um próximo período (o limite de tentativas é também configurável) ou pode-se gerar uma resposta de falha para as camadas superiores, indicando uma solicitação de novos comandos.

Nas subseções a seguir serão descritas cada etapa deste novo protocolo proposto (DyTEE), de forma que este possa operar de maneira robusta e eficiente, visando a economia de energia como um fundamental requisito operacional em sistemas específicos de automação.

3.1 Parâmetros configuráveis do DyTEE

Semelhante ao S-MAC, o novo protocolo emprega a utilização de um único canal de comunicação, a abordagem de sensoriamento da portadora como método de prevenção de colisões e a comunicação em grupos de nós (*clusters*) que estão associados entre si por meio de um cronograma de ativação (*duty cycle*). No entanto, análogo ao TDMA, existe um nó (coordenador) responsável pela coordenação de tempos e sincronização (por meio de pequenos pacotes de controle).

O protocolo DyTEE é composto de vários parâmetros que permitem ao usuário configurá-lo para melhor se adequar à aplicação, sendo alguns deles específicos para toda a rede (PR - parâmetros da rede), como:

- **Número máximo de nós em cada grupo:** Selecionar um número máximo de nós em cada grupo de forma que seja o mais próximo do que será utilizado, no caso de uma rede bem determinada, permite que o protocolo opere de forma ótima. Entretanto, escolher um valor acima não irá prejudicar o funcionamento deste, somente não irá alcançar a economia máxima possível. Existe ainda a possibilidade de trabalhar sem limites de nós, porém isso pode permitir um maior atraso no envio de mensagens, visto o limite de transmissões por período detalhado na Subseção 3.4.5;
- **Ciclo de trabalho máximo:** Determina o tempo máximo que os nós poderão ficar ativos (incluindo sincronização, pacotes de controle e dados úteis) em relação ao tempo em que ficarão inativos;
- **Tamanho máximo de uma mensagem enviada no *beacon*:** No DyTEE, semelhante ao IEEE 802.15.4, faz-se possível a junção de carga útil na mensagens durante os sinais de sincronização. Estas mensagens são configuradas e alteradas em tempo de execução pela própria camada de aplicação;
- **Quantidade de *slots* para registro:** Determina o número de vagas (espaço de tempo) que ficarão disponíveis para registro de novos nós em cada período. A escolha de valores muito pequenos pode dificultar (por colisões) a conexão de novos nós, bem como a escolha de valor muito grandes aumenta o consumo com períodos ociosos;

- **Máximo número de períodos de recuo (em inglês *backoff*):** Neste parâmetro, também chamado em inglês de *backoff*, determina-se quantos períodos um nó pode atrasar sua requisição, ainda que de forma aleatória, para realizar novamente uma tentativa de registro com determinado grupo;
- **Máximo número de tentativas de registro:** Um nó somente irá tentar se registrar por este número de tentativas, após isto é feita a comunicação à camada de aplicação da falha de registro. Porém, caso desejado, a aplicação pode reiniciar este processo por quantas vezes for necessário.
- **Máximo de requisições de transmissões por períodos:** Em aplicações que possuam maior taxa de geração de mensagens, pode-se reduzir o consumo da rede por meio deste parâmetro. Isso pois, sabendo-se que há um grande número de requisições por tempo, torna-se possível selecionar este parâmetro, em paralelo com o ciclo de trabalho máximo, de forma que cada nó realiza em média mais de uma requisição por período. Configurando corretamente estes dois, pode-se reduzir também o total de cabeçalho utilizado no protocolo e consequentemente o consumo também.
- **Cargá útil máxima por mensagem (*payload*):** Cada nó somente poderá enviar esta quantidade de bytes por mensagem. Se a camada de aplicação tiver disponibilidade de acumular mais dados para enviar de uma só vez, por meio deste parâmetro, pode-se reduzir também o consumo de energia da rede.
- **Máximo de mensagens armazenadas:** Devido ao limite de memória existente nos módulos reais, deve-se determinar uma quantidade razoável máxima de mensagens que poderão ficar armazenadas na camada MAC em tentativa de retransmissão. Esse processo de armazenamento é necessário caso a camada de aplicação gere mais mensagens do que o protocolo é capaz de transmitir por tempo. Entretanto este caso de acúmulo deve ser estritamente temporário (dependendo da memória do sistema e da taxa de geração de mensagens), caso contrário haverá perdas constantes de pacotes. Isso ocorrerá sempre que o ciclo de trabalho e/ou número de requisições por período for configurado insuficientemente para determinada aplicação.
- **Máximo número de tentativas de transmissão de mensagens:** No DyTEE, as falhas de transmissão só devem ocorrer por falhas na recepção e não por colisão. Sendo assim, este parâmetro permite que a camada MAC responda à camada de aplicação com um resultado positivo ou negativo a respeito da requisição de transmissão de uma mensagem.

Existem ainda os parâmetros específicos de cada módulo (PM - parâmetros do módulo), determinados em tempo de programação do próprio. Isso pois, ao solicitar à camada MAC uma conexão à um grupo, a função de solicitação exige estes parâmetros, os quais são:

- **Canais disponíveis:** Os nós somente irão procurar ou gerar redes nos canais disponíveis por este parâmetro.

- **Função exercida:** É possível forçar que determinado nó exerça uma função desejada, como coordenador ou folha (nó ponta, do inglês *end-device*). Dessa forma, caso seja solicitado que determinado nó seja coordenador, este irá gerar uma rede em algum dos canais disponíveis que possuir menor leitura de ocupação (detecção de energia ou CCA), independente se já houver outro grupo operando neste canal. Entretanto este irá procurar um período livre, dado que outro grupo já tenha escolhido um ciclo de ativação. Por isso a necessidade de um ciclo de ativação máximo, dado pelos parâmetros anteriores. No caso de um nó ser determinado pela aplicação como do tipo folha, este irá sempre procurar grupos e nunca iniciar o seu próprio.
- **Identidade original:** Indica à camada MAC qual a identidade que este nó deseja utilizar para se registrar. Em um conjunto de nós, pode-se aceitar que um ou mais nós possuam as mesmas identidades sem prejuízo ao sistema, contanto que estes não estejam operando no raio de duas vezes o de alcance um do outro. Isso pois, estes nós irão adquirir novas identidades após se registrarem em um grupo, resolvendo este problema. Por padrão, se utiliza dois bytes de identidade por grupo.

Os PM também podem ser determinados como genéricos pela camada de aplicação, ou seja, uma terceira opção que permite ao módulo utilizar qualquer canal disponível, identidade aleatória e função completa (pode virar coordenador ou folha). Assim, no caso dos canais, tenta-se utilizar algum dos disponíveis na tecnologia. No caso da função exercida, este módulo irá realizar o processo padrão do DyTEE de inicialização de uma rede descrito posteriormente. No caso da identidade, gera-se um valor aleatório deste, porém se recomenda que seja atribuído um valor fixo e individual para cada nó.

Quando um nó tenta se conectar à rede com um endereço de grupo (PanID do inglês *personal area network identification*) já determinado, por exemplo 0xBE, este possui o primeiro byte da esquerda sempre igual a zero, de forma a indicar que ele não pertence àquela rede ainda, enquanto o segundo byte é determinado pela aplicação, por exemplo o valor hexadecimal 0x00xx. Quando o registro é efetivado, o novo nó recebe um novo endereço relativo ao grupo e o seu primeiro byte se torna igual à identidade daquele grupo, dessa forma o novo endereço deste nó seria 0xBEyy. Exceto no caso de criação de grupo, onde o nó criador (coordenador) também altera seu endereço de forma que o grupo possua seu endereço anterior e seu novo endereço dentro do grupo será igual a zero (por ser o coordenador), como no exemplo em que o nó inicia com endereço 0x00BE e, após criar o grupo, assume a identidade de 0xBE00, onde seu grupo possuirá o PanID 0xBE.

É importante citar que certos parâmetros de criação da rede, como o número máximo de nós e o tempo máximo ativo descritos acima, estão alterados de sua forma de determinação segundo a proposta inicial do protocolo [38]. Isso ocorreu pois durante o desenvolvimento do protocolo no dispositivo físico, foram percebidas outras necessidades que não eram visíveis nas fases iniciais da construção do protocolo. Entretanto, essas mudanças não causaram significativas diferenças nos resultados, somente na forma como o protocolo é configurado. Contudo, o DyTEE descrito neste trabalho segue de acordo com o que foi implementado em módulo real, sendo, portanto, factível e funcional.

O valor determinado para simular as aplicações desejadas será exposto nos capítulos seguintes, onde serão apresentados também os resultados das simulações e implementações reais.

3.2 Início de uma rede

Para iniciar uma rede, os nós procuram por um sinal de sincronização (em inglês *beacon*) de um coordenador, de forma que se realize uma conexão (explicada nas subseções seguintes) e se junte a esta rede existente. Uma vez que um nó recebe esse sinal de sincronismo, trocam-se informações com o coordenador para então se obter um número de identificação para comunicação com esse grupo específico. Entretanto, se não houver nenhum grupo formado, o primeiro nó a iniciar naquele ambiente irá tentar formar o seu próprio grupo, enviando periodicamente sinais de *beacons* (processo semelhante ao S-MAC).

Para a comunicação dos nós em múltiplos saltos, formando um caminho de dados, os módulos já conectados em um coordenador podem continuar procurando por novos grupos periodicamente, a fim de manter as informações de conexão disponíveis para formar novas redes e expandir suas rotas.

O processo padrão para que um nó comece a enviar sinais de sincronismo baseia-se em um período de espera para averiguar se já não existem outras redes ao alcance, seguido por um intervalo aleatório de espera baseado na quantidade percentual de energia disponível para esse nó (presume-se que o nó seja capaz de medir sua própria carga de energia disponível). Quanto maior a energia disponível (100% de carga até o mínimo possível operacional da bateria), mais rápido o nó tentará se tornar o líder do grupo. No entanto, é importante notar que, mesmo se o nó possuir 100% de carga disponível, o tempo de espera para o envio do primeiro sinal ainda será aleatória, mas provavelmente isso aconteça antes de um nó com menor carga da bateria.

A coordenação de um grupo alterna entre nós de acordo com a porcentagem de utilização da bateria. Assim, quando um líder utiliza uma quantidade predefinida de energia, este anuncia o fim do grupo. Assim, os módulos, agora desassociados e sem grupo, iniciam uma nova competição pela liderança. Este procedimento evita o uso excessivo da bateria de um único nó, proporcionando consumo aproximadamente igual de energia entre nós. Neste processo, pode-se ocorrer o surgimento de mais de um grupo, nos casos onde o próximo líder não possua alcance com todos os outros nós do grupo recém finalizado.

3.3 Estrutura geral do protocolo

Uma vez estabelecida a rede, os nós irão seguir um fluxo de operações para realizar suas comunicações, isto pode ser visto na Figura 3.1, um diagrama simplificado de eventos temporais do DyTEE. Como pode ser visto, logo após o sinal de sincronização (*beacon*, o qual possui períodos determinados nos parâmetros da rede descritos anteriormente), o protocolo realiza um momento de organização da rede (SyncM) seguido por ciclos de ativações e transmissões. Durante o SyncM,

cada nó conectado nesta rede irá tentar anunciar para o coordenador sua requisição de transmissão que informa o destino pretendido para o envio de seus pacotes de dados. As requisições são feitas por meio de vagas (*slots* de tempo) destinadas para cada nó conectado, não havendo então colisão neste processo.

Todo o período que um nó possui para enviar dados e receber as confirmações (ACK) é indicado na Figura 3.1 por PKT. Assim, um nó apenas enviará dados, se ele obtiver sucesso em informar sua requisição durante o SyncM. Este sucesso irá ser confirmado durante o fim do SyncM, onde o coordenador envia para todos os nós de uma só vez (em inglês *broadcast*) uma mensagem com as agendas que serão respeitadas nos períodos de ativação seguintes. Esta agenda contém a ordem e os pares que irão se comunicar em cada vaga disponível (PKT). É importante notar que estas vagas são separadas por espaços vazios de tempos pequenos o suficiente para evitar problemas com desvios de relógio (em inglês *clock drift*), que de acordo com [19], se os nós forem sincronizados em períodos de até 10 segundos, a diferença entre os relógios dos nós não deve exceder 0,2 milissegundos por segundo, por exemplo.

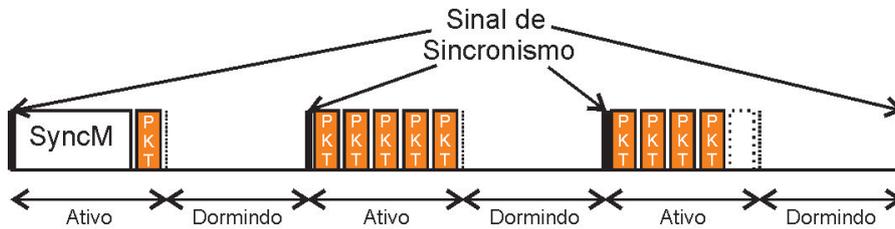


Figura 3.1: Diagrama simplificado de eventos temporais do protocolo DyTEE.

Ao contrário do S-MAC, no DyTEE não se permite que qualquer nó dentro de um grupo possa estar ativo por mais tempo do que o período ativo determinado pelo ciclo de trabalho. Portanto, se existirem outros grupos (*clusters*) nas proximidades, esse comportamento evitará interferências no funcionamento entre eles. Desta forma é possível que vários grupos possam operar simultaneamente, contanto que estejam a uma distância mínima de $2r$, onde r é o raio de alcance.

No momento de sincronização pode se disponibilizar um tempo máximo dado pelo período máximo de ativação que cada período pode ter. Em outras palavras, se o período total (ativo + dormência) do grupo é de t segundos, com um ciclo de trabalho de 10%, o SyncM poderia durar no máximo 0,1 segundos de t .

No entanto, dados os parâmetros da rede, o tempo máximo necessário para realizar o SyncM (T_{SyncM}) pode ser ainda menor, baseado no número de nós conectados ao grupo. Assim sendo, após o fim do SyncM, caso ainda haja tempo suficiente para transmissão de dados, os nós podem começar a enviar seus pacotes imediatamente, respeitando o tempo máximo de ativo, como pode ser visto no exemplo da Figura 3.1.

Dado que o tamanho máximo dos pacotes é determinado em tempo de programação, a duração $T_{Transmissão}$ para o envio de um pacote é constante (em torno de 5,7 ms, dados da implementação realizada). Assim, possuindo o valor do *período máximo de ativação*, o *tamanho máximo da*

mensagem e o *tamanho máximo das mensagens transmitidas durante um sinal*, pode-se determinar o *número máximo de transmissões de dados* que ocorrerão durante cada período, dada a equação

$$Max_{transmissões} = \left\lfloor \frac{(T_{Máx(Ativo)} - T_{Máx(Beacon)})}{T_{Transmissão}} \right\rfloor, \quad (3.1)$$

onde $Max_{Transmissões}$ é o valor máximo (arredondado para baixo) de transmissões que ocorrerão em um período, $T_{max(Ativo)}$ é o tempo máximo ativo dado pelo ciclo máximo de trabalho e $T_{max(Beacon)}$ é o tempo máximo necessário para a realização dos sinais de sincronização (*beacons*), os quais serão explicados detalhadamente nas subseções seguintes.

Antes de tudo, é importante citar que todas as mensagens neste protocolo são precedidas de um cabeçalho fixo, pois sem este o receptor não seria capaz de decifrar o significado de cada mensagem captada. Bem como existe a necessidade de distinguir, no cabeçalho, se a mensagem recebida pertencem ao próprio protocolo utilizado. Isso pois, dado que este foi implementado na mesma plataforma que executa o IEEE 802.15.4, é cabível que em um ambiente cheio de outros aparelhos funcionando, como roteadores IEEE 802.11 e outros módulos IEEE 802.15.4, possa haver interferências e recepções de mensagens que não fazem parte de uma rede com o protocolo aqui proposto.

Com isso, o cabeçalho fixo é dado pelo primeiro byte sendo sempre o total de bytes que deverão ser lidos em seguida, enquanto o segundo corresponde ao campo de controle do pacote (FCF do inglês *frame control field*) e o terceiro e último byte fixo corresponde somente ao termo definindo o protocolo DyTEE (no caso, em hexadecimal, 0xDE). Os valores de FCF podem ser dados como:

- **FCF_FRAMETYPE_SIMPLE_BEACON:** Define o pacote como BS simples, ou seja, após este devem ocorrer transmissões de mensagens e não do SyncM;
- **FCF_FRAMETYPE_SYNCM_BEACON:** Define o pacote como BS para SyncM, ou seja, após este deve ocorrer um momento de sincronização da rede;
- **FCF_FRAMETYPE_DATA:** Define o pacote como PKT, ou seja, uma mensagem de um nó para outro transmitindo dados úteis;
- **FCF_FRAMETYPE_ACK:** Define o pacote como ACK, confirmando os dados recebidos em uma comunicação;
- **FCF_FRAMETYPE_REGISTER_REQUEST:** Define o pacote como uma requisição de registro à rede;
- **FCF_FRAMETYPE_DISASSOCIATE:** Define o pacote como uma requisição de desassociação da rede;
- **FCF_FRAMETYPE_REQUEST_SLOT:** Define o pacote como uma requisição de transmissão, onde o nó pertencente à rede informa ao coordenador que deseja transmitir algo para outro nó;

- **FCF_FRAMETYPE_MAC_CMD:** Define o pacote como sendo uma mensagem *broadcast* enviada pelo coordenador para agendar os eventos do grupo.

Existem ainda informações que sempre são adicionadas ao final dos pacotes transmitidos, de forma a verificar se a transmissão foi correta (FCS do inglês *frame check sequence*). Ao verificar a recepção da mensagem corretamente, essa informação de FCS é descartada e substituída por parâmetros medidos do canal, como o indicador de qualidade da conexão (LQI do inglês *link quality indication*) e o nível de energia (ED), os quais são encaminhados para as camadas de comunicação superiores. Alguns desses parâmetros serão melhor descritos posteriormente neste texto.

3.4 Estrutura do momento de sincronização

O momento de sincronização (SyncM) possui quatro subdivisões, como pode ser visto na Figura 3.2.



Figura 3.2: Diagrama temporal de um momento de sincronização (SyncM).

Cada divisão é composta por nós transmitindo ou recebendo do coordenador, segundo a topologia estrela (revisada na Subseção 2.2.2). Entre estes instantes, ocorre o principal deles, chamado de *broadcast*, pois nele é feita a divulgação de toda a agenda que deverá ocorrer no grupo nos próximos períodos, até que ocorra novamente a transmissão de outro SyncM.

3.4.1 Sinal de sincronização - BS (*Beacon Signal*)

No início de cada período e também do SyncM, ocorre o envio de um sinal nomeado aqui como BS (do inglês *beacon signal*), onde há a transmissão de um pacote enviado pelo líder do grupo, utilizado para atualizar os relógios de todos os nós associados ao grupo. O relógio deste nó é contado relativamente, ou seja, com base nestes momentos eles irão contar o tempo para todas as ações seguintes tendo como marco zero esta recepção, até a ocorrência do próximo BS. O pacote de BS contém a seguinte combinação de bytes indicada na Tabela 3.1.

Tabela 3.1: Sequência de bytes do pacote de controle BS.

1º byte	2º byte	3º byte	4º byte
Comprimento da mensagem em bytes (n)	FCF	Identificador DyTEE	Identidade do grupo
5º byte	6º byte ...	(n-1) byte	n byte
Número de nós conectados	Mensagem com carga útil	FCS	FCS

Como pode ser visto na Tabela 3.1, além dos bytes fixos de cabeçalho e dos bytes de FCS, no BS são transmitidos também o número de nós conectados no grupo, a identidade do grupo e os bytes de mensagens determinada pela aplicação.

Por meio do número de nós conectados, novos módulos que procurem se conectar com este grupo saberão se existe ou não espaço disponível neste grupo, tendo em vista o parâmetro de rede (PR) determinado. O número de nós conectados também serve de determinação para indicar quando ocorrerá cada evento seguinte dentro do SyncM, ou seja, sabe-se quanto tempo irá durar o período de requisições de transmissões e conseqüentemente quando ocorrerá a transmissão do *broadcast* do SyncM.

É importante citar que o tempo de transmissão de cada byte possui valor fixo determinado relativamente à tecnologia utilizada. Considerando a tecnologia 2.4GHz e o protocolo IEEE 802.15.4, cada byte demora 32 microssegundos e com essa estimativa, pode-se subtrair a quantidade de bytes enviados de modo a determinar o momento em que se iniciou a recepção dos dados, isso desconsiderando o atraso de propagação.

Os sinais de *beacon* introduzem tanto os períodos em que ocorrem sincronização da rede (dado pelo FCF com SyncM), como os períodos que ocorrem somente transmissão de dados (dado pelo FCF simples). A determinação de qual BS será transmitido pelo coordenador é uma função dada após o final do *broadcast* e será melhor explicado à frente.

Ao final da transmissão do BS, há sempre um espaçamento de tempo (t_{Esp}) necessário para evitar problemas com desvio do relógio (sincronismo) e também para que o módulo consiga decifrar e atuar com base no pacote recebido, ou seja, o tempo em que o *hardware* necessita para operar sobre esta situação.

3.4.2 Período de registro de novos nós - REG

O processo de registro em um grupo é feito após o momento em que um dos nós na rede já tenha iniciado os envios de BS. Se outro módulo deseja se conectar ao grupo, depois de receber o BS referente ao SyncM, mesmo estando esse módulo em outro grupo ou não, este pode realizar um processo de registro com o nó líder deste novo grupo durante período REG deste. Este processo é dado por tentativas em um processo semelhante ao protocolo CSMA com vagas (em inglês *slotted CSMA*).

O método de requisição de registro por CSMA com vagas foi escolhido para esta parte do protocolo devido à sua simplicidade, no qual um nó envia um pacote de solicitação para registro (RTR do inglês *request to register*) e receberá sua confirmação de registro (RACK do inglês *register acknowledgment*) do líder, enviado juntamente com demais informações no *broadcast* do SyncM. Dentro deste próximo *broadcast* será informando o novo número de identificação (ID) para este novo nó referente àquele grupo. No entanto, uma vez que é um método semelhante ao CSMA, podem haver colisões nas solicitações de registro ao grupo¹, que somente causa um atraso neste

¹Note que as únicas colisões possíveis no DyTEE são durante os processos de registro e com os pacotes de requisição de registro no período REG (contenção).

processo de registo.

Este período REG e suas possíveis colisões não afetam as transmissões de nós já registrados no grupo, só em nós que estão esperando para registrar neste grupo também. De fato, as únicas colisões esperadas no DyTEE são devido a novos registros nos grupos, isso considerando ainda o fato de serem realizados procedimentos de detecção de colisão (CCA).

Contudo, caso ainda haja uma colisão no registro, a nova tentativa de associação de um nó que não obteve resposta durante o *broadcast* acontece durante uma vaga aleatória no intervalo REG de algum SyncM posterior aleatório, ou seja, não necessariamente no próximo ciclo. Sendo a distribuição de probabilidade uniforme para escolher a vaga REG e os parâmetros de recuo determinados de acordo com a aplicação (parâmetros da rede). A Figura 3.3 apresenta um fluxograma do processo de registo.

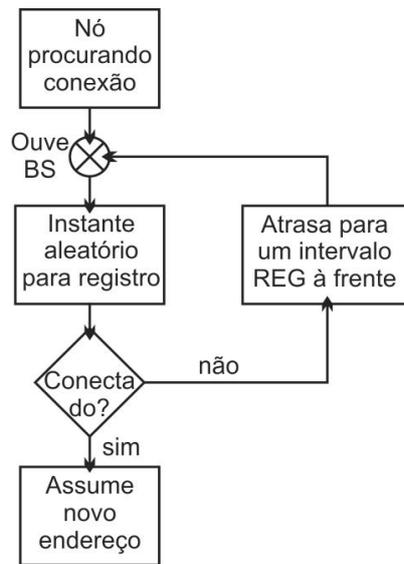


Figura 3.3: Fluxograma de registo na fase REG do SyncM.

Quando um nó envia a solicitação de registo (RTR) com sucesso, o líder do grupo analisa a possibilidade de registo e então insere a resposta juntamente ao pacote enviado durante o *broadcast* do SyncM. Esta pode conter um valor negativo (seja por grupo estar cheio, por motivos internos ou por falha na recepção) ou positivo. A resposta positiva contém o ID para este nó neste novo grupo.

O pacote de solicitação de registo enviado durante o intervalo REG contém 6 bytes, como apresentado na Tabela 3.2.

Tabela 3.2: Sequência de bytes do pacote de requisição de registo.

1º byte	2º byte	3º byte	4º byte	5º byte	6º byte
Comprimento da mensagem em bytes (n)	FCF	Identificador DyTEE	Identidade do nó	FCS	FCS

Semelhante ao evento de BS, após o período REG ocorre também um espaçamento de tempo (t_{Esp}) por questões de *hardware* e sincronismo.

3.4.3 Requisição de vagas de transmissão - RSlot

Os espaços de tempo do pacote de requisição de vagas (RSlot do inglês *request slot*) seguem logo após o intervalo REG e é sempre arranjado em uma sequência numérica ascendente que varia de um até o número de nós registrado naquele momento. Desta forma, cada nó conectado ao grupo (ID único no grupo) pode solicitar ao líder, em sua respectiva vaga (RSlot) correspondente a sua identificação, uma vaga de transmissão. Dessa forma este nó poderá enviar uma mensagem de dados para outro nó deste grupo.

Um RSlot tem uma duração de tempo tal que seja possível enviar suas informações de ID, os IDs dos nós aos quais este deseja transmitir e uma margem de tempo de espaçamento. Assim, o RSlot fica como na Tabela 3.3.

Tabela 3.3: Sequência de bytes do pacote de controle RSlot.

1º byte	2º byte	3º byte	4º byte
Comprimento da mensagem em bytes (n)	FCF	Identificador DyTEE	Identidade do nó neste grupo
5º byte	6º byte ...	(n-1) byte	n byte
PanID do grupo	Identidade dos nós a transmitir	FCS	FCS

Qualquer um dos participantes do grupo que não tenha mensagem para enviar, deve ainda enviar um RSlot ao coordenador informando sua presença na rede, pois em casos onde, por algum motivo, o nó desconecte, torna-se possível, por esse sistema de RSlot, detectar a ausência de um nó e conseqüentemente excluí-lo do grupo, liberando espaço e reduzindo desperdícios. Por meio do RSlot, também é possível requisitar uma saída (exclusão) do grupo, seja porque o nó sabe que irá desligar definitivamente ou porque este deseja somente não participar de determinado grupo.

Na Figura 3.2 são mostradas cinco vagas de requisição de transmissão (região verde). No entanto, este não é necessariamente o número máximo de nós que podem ser conectados neste grupo. O valor máximo de nós em um grupo é dado por outra equação em função dos PR e será explicada mais a frente.

3.4.4 Divulgação da agenda - *Broadcast*

Após o envio do BS, período de registro (REG) e período para envio de requisições de transmissão (RSlot), vem o último evento do SyncM, a divulgação da agenda do grupo (*Broadcast* enviada pelo líder).

Nesta transmissão, o nó coordenador informa para todo o grupo em uma só mensagem o resultado das operações ocorridas durante o SyncM, ou seja, quais nós se conectaram, quais desconectaram e quais pares do grupo irão comunicar, dada a sequência em que o líder organizar. Para proporcionar uma maior justiça na ordem de transmissão dos nós, esta sequência dos pares de transmissão enviada pelo coordenador é embaralhada de forma aleatória. Assim, o pacote enviado no *broadcast* é constituído segundo o descrito na Tabela 3.4.

Tabela 3.4: Sequência de bytes do pacote de controle enviados no *broadcast*.

1º byte	2º byte	3º byte	4º byte	(4 + 2NR) byte
Comprimento da mensagem em bytes (n)	FCF	Identificador DyTEE	Número de respostas de registro (NR)	Antigo endereço e novo endereço para este grupo (em pares)
(4 + 2NR + 1) byte	(4 + 2NR + 1 + 2ND) byte	(4 + 2NR + 1 + 2ND + 1) byte ...	(n-1) byte	n byte
Número de nós desassociando (ND)	Endereço no grupo e razão para saída (em pares)	Pares de nós comunicantes nas próximas vagas	FCS	FCS

Como pode ser visto na Tabela 3.4, o cabeçalho mínimo do pacote de *broadcast* é dado quando não há nenhuma requisição de transmissão, nem registro (NR) ou desassociação (ND), tendo assim o mínimo de cabeçalho do protocolo (S_{Cab}), cinco, mais dois bytes (indicando zero de NR e zero de ND), totalizando o mínimo de sete bytes de controle para o envio deste pacote.

As respostas de registro são enviadas de par em par, ou seja, os nós recebem no primeiro byte após o NR, o valor do antigo endereço que um nó fora da rede possuía e no byte seguinte, o valor do novo endereço que este nó irá possuir dentro deste grupo, ou seja, um valor combinado com a identidade PanID deste grupo.

No caso das respostas de desassociação, são somente enviados os endereços dos nós que estão dentro deste grupo e serão desconectados, seguidos do byte que indica a razão pelo qual este nó sairá. Esta razão pode ser dada pelo próprio coordenador ou pela requisição do nó do grupo (feita durante o RSlot). A necessidade de conservação da bateria do líder, pode fazer com que este encerre o grupo após determinada percentagem de consumo, como explicado anteriormente, desta forma, o anúncio de desassociação contendo o endereço do próprio líder permite aos nós da rede entenderem que o grupo será fechado.

Desta forma, o tempo de transmissão do *broadcast* juntamente com o espaçamento (t_{Esp}) para interpretação dos dados e a necessidade de evitar desvios dos relógios tem uma duração máxima ($T_{Broadcast}$) dada por

$$T_{Broadcast} = \frac{S_{Broadcast}}{TX_{TAXA}} + t_{Esp}, \quad (3.2)$$

onde a taxa de transmissão do rádio (TX_{TAXA}) é dada em kilobytes por segundo e $S_{Broadcast}$ é o tamanho, em bytes, do pacote de transmissão enviado para todos os nós. Este é igual à soma do tamanho do cabeçalho (S_{Cab}), incluindo as informações, respostas de registros e comandos que o líder irá transmitir, mais o número de nós requisitando transmissão no grupo ($Nós_{req}$) vezes o tamanho do par de endereços (S_{Pares} , onde no caso da implementação atual, o par é constituído de somente dois bytes), que representa os nós que irão se comunicar nas vagas seguintes, de acordo com a sequência programada para transmissão de dados. Por conseguinte, resulta que

$$S_{Broadcast} = S_{Cab} + 2 + 2(NR + ND) + (Nós_{req})S_{Pares}. \quad (3.3)$$

Como o padrão utilizado no DyTEE é de dois bytes de endereço, sendo o primeiro (byte de maior significância) relativo ao PanID do grupo e o segundo relativo ao próprio endereço dele dentro desta rede, para indicar os pares de comunicação, o coordenador somente precisa de dois bytes, um indicando qual irá transmitir e o outro indicando qual irá receber para determinada vaga.

Os valores máximos de NR e ND são dados pelo PR quantidade de vagas para registro, pois não podem haver mais respostas do que requisições. Dessa forma, se houver cinco vagas como parâmetro de quantidade de vagas para registro, então NR e ND no máximo terão valor cinco.

3.4.5 Parâmetros dependentes

Além dos parâmetros determinados pelo usuário, como os PR e PM citados anteriormente, existem ainda os parâmetros que são consequências destes, tais como:

- **Limite do valor máximo de nós ($Nós_{Máx}$) disponível para a configuração do PR;**
- **Duração máxima de uma transmissão de dados;**
- **Duração máxima de um sinal de sincronização (BS);**
- **Máximo de transmissões por período (equação 3.1).**

Durante a escolha dos valores de PR e PM e compilação do protocolo, existe uma verificação das combinações destes valores, de forma a avaliar a compatibilidade deles. Desta forma, evita-se resultados indesejados e falhas no sistema, como é o caso da quantidade máxima de bytes em uma mensagem. O *hardware* atualmente utilizado nesta implementação foi o mesmo que atende às especificações do IEEE 802.15.4 [11], portanto o número máximo de bytes transmitidos em um só pacote é de 127 bytes. Assim, o máximo de informação dentro de um pacote de *broadcast* do SyncM é limitado, ou seja, existem finitos pares de bytes de transmissões (vagas) a serem divulgadas em uma só mensagem. Assim, o valor máximo de nós em um grupo que pode ser configurado nos parâmetros de rede (PR) é dado por

$$Nós_{Máx} = \left\lfloor \frac{127 - (S_{Cab} + 2 + 2NR_{Máx} + 2ND_{Máx})}{S_{Pares}} \right\rfloor, \quad (3.4)$$

onde $NR_{Máx}$ e $ND_{Máx}$ são iguais ao PR selecionado, isto é, quantidade de vagas para registro.

A escolha de um valor de PR, número máximo de nós em cada grupo, maior que $Nós_{Máx}$ irá subentender que o protocolo não irá respeitar $Nós_{Máx}$. Nestes casos, existem duas soluções:

- **Utilizar mais de uma mensagem de *broadcast*:** Com o acréscimo de informação no cabeçalho do *broadcast* é possível que se transmita mais de uma mensagem, de forma sequencial. Assim, a partir de a segunda mensagem em diante do *broadcast* será somente uma continuação do primeiro.
- **Bloquear transmissões aleatoriamente:** Nesta opção utiliza-se o valor de $Nós_{Máx}$ para transmitir o máximo de vagas possível e, de forma aleatória, selecionar os pares que irão se comunicar e deixar de fora, porém na fila, os nós que não forem selecionados. Esta opção é mais recomendada para redes que não se saiba qual número máximo de nós a ser utilizado, bem como em redes que possuam uma variação grande de frequência de requisições, com média abaixo de $Nós_{Máx}$.

Outra verificação importante é da soma dos períodos dentro do SyncM, onde BS, REG, RSlot vezes o número máximo de nós no grupo e tempo de transmissão do *broadcast* não podem exceder o tempo máximo de ativo determinado pelo PR do grupo, ciclo de trabalho (*duty cycle*) máximo, então

$$T_{Ativo\ máx} \geq T_{BS} + T_{REG} + Nós_{Máx} \cdot T_{RSlot} + T_{Broadcast}. \quad (3.5)$$

Como esses parâmetros são os mesmos para todos os nós, após a sincronização do relógio dos nós durante o período de BS, todos os membros do grupo sabem exatamente quando ocorrerá cada outro momento do protocolo, ou seja, o momento em que cada nó se comunicará. Então o líder só precisa transmitir a agenda de eventos e a quantidade de nós na rede para que todos os nós se organizem sem colisão.

3.5 Transmissão de pacotes de dados

Dado que cada membro do grupo recebe durante o SyncM o número total de nós conectados naquele momento, ao final do *broadcast* é possível determinar se haverá tempo suficiente para a comunicação entre o primeiro par de nós determinado pelo líder, bem como se haverá tempo suficiente para as seguintes sequências de transmissão até o fim do período ativo. No final do período ativo, todos os membros do grupo irão dormir e esperar para o próximo ciclo de ativação, reduzindo o consumo de energia, uma vez que a energia consumida da bateria durante a período de dormência é muito menor do que a energia consumida no período ativo. Quando se inicia o próximo período, os nós voltam a seguir a agenda de transmissão recebida anteriormente (no último *broadcast* de SyncM). Isto acontece até o fim de toda a sequência determinada no *broadcast* do SyncM, até que seja feito um novo SyncM, recomeçando tudo novamente.

A estrutura do pacote de dados é apresentada na Tabela 3.5.

Tabela 3.5: Sequência de bytes do pacote de dados.

1º byte	2º byte	3º byte	4º byte	5º byte
Comprimento da mensagem em bytes (n)	FCF	Identificador DyTEE	Número sequencial de dados (DSN)	PanID do nó receptor
6º byte	7º byte ...	(n-1) byte	n byte	
Endereço do nó receptor	Dados úteis	FCS	FCS	

No DyTEE é enviado somente o endereço do nó receptor, bem como o DSN (do inglês *data sequence number*) para se evitar que outros nós possam, de fora deste grupo, ocasionalmente ouvir esta mensagem e não saberem que não se destina a eles, ou seja, garantir que somente o receptor irá decifrar esta mensagem. Entretanto, caso se garantisse a criação de somente um grupo na rede, não seria necessário informar estes valores, pois durante o *broadcast* do SyncM, os nós já sabem e guardam de quem irá receber determinada mensagem em determinada vaga, bem como quem deverá enviar. De fato, o processo de guardar esses valor só interessa aos nós que irão participar da transmissão em tal vaga.

De forma semelhante, o pacote de confirmação (Ack) possui os seguintes bytes especificados na Tabela 3.6.

Tabela 3.6: Sequência de bytes do pacote de controle ACK.

1º byte	2º byte	3º byte	4º byte	5º byte	6º byte
Comprimento da mensagem em bytes (n)	FCF	Identificador DyTEE	DSN recebido	FCS	FCS

A duração máxima de uma transmissão de dados é dada pelo tempo de espaçamento (t_{Esp}), tempo de transmissão dos dados (T_{dados}), tempo de recepção e interpretação do receptor ($T_{Int REC}$), tempo de resposta de confirmação (ACK), tempo de recepção e interpretação do primeiro transmissor ($T_{Int TRANS}$) e um ultimo espaçamento novamente, ou seja,

$$T_{transmissão} = t_{Esp} + T_{dados} + T_{Int REC} + T_{Ack} + T_{Int TRANS} + T_{Esp}, \quad (3.6)$$

onde T_{dados} é obtido pelo tamanho máximo determinado no PR, carga útil máxima por mensagem, mais o cabeçalho utilizado para dados (Tabela 3.5) dividido pela taxa de transmissão do *hardware* (TX_{TAXA}).

É importante citar também que alguns tempos, como a troca de função entre transmissor e receptor, ligar ou desligar o rádio e outros, já estão considerados dentro dos T_{Int} e t_{Esp} utilizados.

3.6 Descrição do protocolo DyTEE para um nó do grupo

Baseado no protocolo DyTEE detalhado anteriormente, cada nó tem uma sequência definida de tarefas com base nas mensagens transmitidas ao seu redor, o que pode ser expresso como um fluxograma. A Figura 3.4 apresenta cada evento e situação que um nó pode estar operando durante seu funcionamento.

É importante lembrar que cada nó pode tentar tornar-se líder em sua inicialização, mas somente se este estiver programado como coordenador ou modo genérico nos PM e não receber qualquer sinal de sincronização (BS) de outro nó durante sua inicialização. Contudo, ainda que citado anteriormente, mas não apresentado neste fluxograma, o líder do grupo pode, sempre que decidir ser necessário, fechar o grupo. Isso causará uma nova competição entre os módulos dessa rede, de forma a realizarem um novo grupo com outro líder (divisão de consumo para organização de grupos).

3.7 Conclusão do capítulo

Neste capítulo foi descrita detalhadamente cada etapa do processo de funcionamento do novo protocolo DyTEE. Os parâmetros necessários para configurá-lo para cada aplicação desejada, bem como suas restrições.

Pode ser visto que o DyTEE possui um mecanismo sequencial de operação que permite que este realize transmissões sem colisão de dados, mantendo ainda simplicidade em seu funcionamento. Isso sem deixar haver desperdício do canal nos casos onde há nós querendo transmitir, isso pois sua forma dinâmica de organização de vagas consome certo tempo com pacotes de controle, porém são compensados pela organização desses *slots* para transmissão não havendo momentos de canal ocioso por colisões ou desencontros dos nós transmissores e receptores.

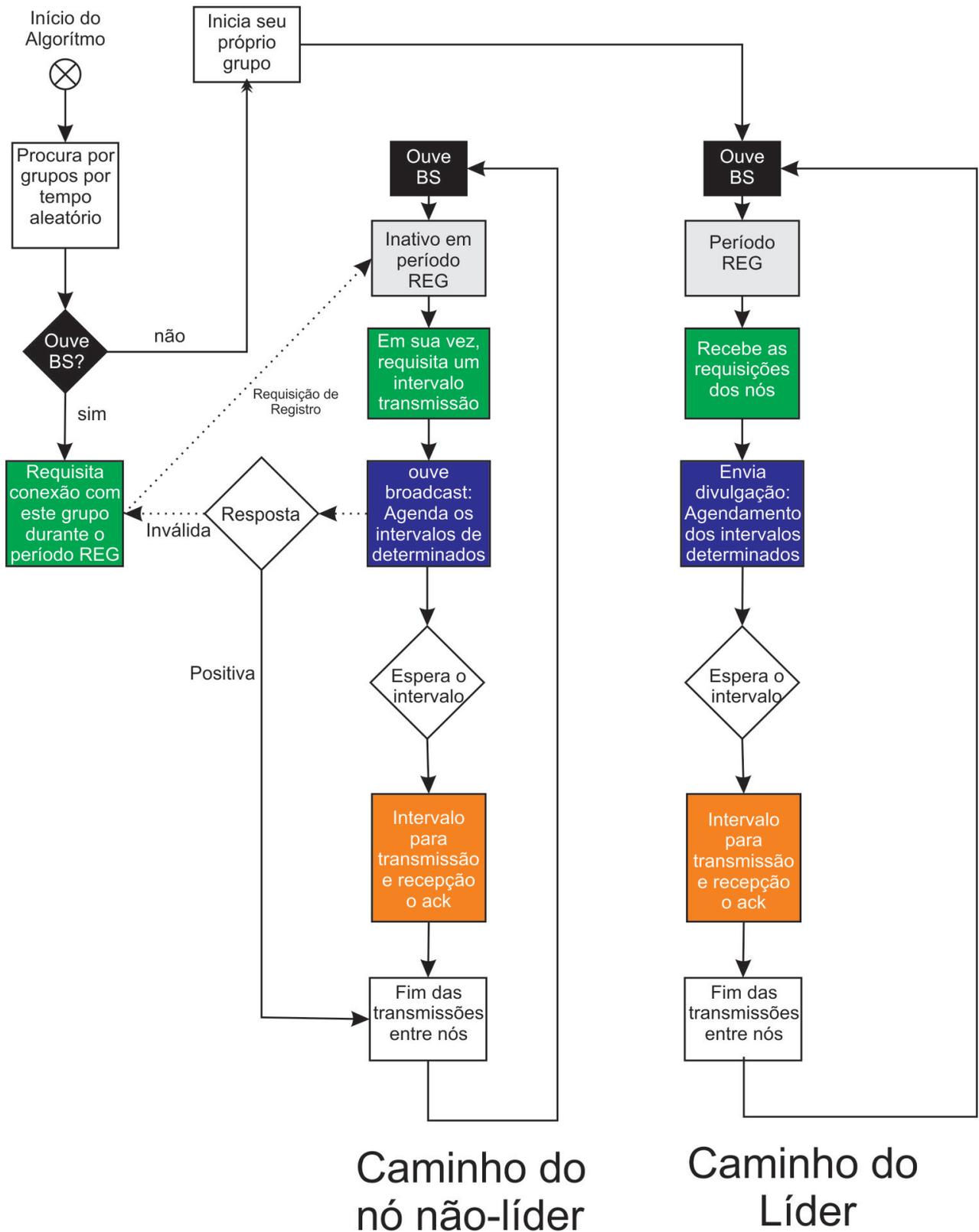


Figura 3.4: Fluxograma de operação para um nó que está executando o protocolo DyTEE.

Capítulo 4

Metodologia

Este capítulo descreve os métodos utilizados neste trabalho para simular e implementar os protocolos em ambientes controlados para comparação de métricas de eficiência, simulando as aplicações desejadas.

Para a implementação do DyTEE, inicialmente foram feitas pesquisas sobre as principais qualidades dos demais protocolos da camada MAC, para que fosse possível observar quais estratégias atingiram melhores resultados visando o tipo de aplicação desejada, ou seja, automação predial e rural. Como citado anteriormente, o fato deste tipo de aplicação poder possuir mensagens muito pequenas, aumenta proporcionalmente o excesso de cabeçalho (*overhead*), uma vez que o tamanho dos pacotes de controle estão próximos do tamanho dos pacotes de mensagem útil. Assim, aumentando também o consumo geral da rede. Por isso, a prioridade do DyTEE é reduzir o gasto energético também pela redução do uso de pacotes de controle.

Por outro lado, para comparar a proposta desenvolvida com os resultados obtidos de outros protocolos, faz-se necessário o uso de métricas específicas, como vazão e latência, entre outras. Dessa forma, foram executadas simulações e testes em módulos reais com objetivo de comparação entre o DyTEE e demais protocolos (no caso S-MAC [10] e IEEE 802.15.4 [11]). A escolha das métricas, parâmetros e topologias foram feitas com base não só nas necessidades deste trabalho, mas também nas restrições existentes e são melhor discutidas na conclusão deste capítulo.

4.1 Desenvolvimento e simulações

A primeira etapa de comparação foi feita por meio de simulações de eventos criados por roteiros de execução no *software* MATLAB [9]. Nesta, foram implementados tanto o S-MAC mais simples [10], quanto o próprio DyTEE. No entanto, no artigo do S-MAC foram utilizadas tecnologias de *hardware* mais antigas, como a plataforma MICA [39], assim os parâmetros deste módulo, como taxa de transmissão, valores de consumo dos estados (ativo, dormência, ocioso e de recepção) e outros, são bastante diferentes, bem como o tempo em que estes módulos levam para interpretar suas mensagens recebidas. Dessa forma, nesta primeira fase de simulação, o protocolo DyTEE foi

implementado utilizando esses parâmetros, para que a comparação entre o proposto pelo S-MAC e o DyTEE fossem justas.

A simulação por programação em MATLAB é feita por eventos temporais, ou seja, tem-se na programação um grande laço responsável por simular cada 0,1ms de operação dos módulos na rede, repetindo todo o processo após o fim de um período (ativo e de dormência). Assim, os eventos temporais são marcados por um momento no tempo, em que quando a variável de tempo se equivalesse a este momento, todas as suas consequências são resolvidas, simulando uma operação real.

Os resultados deste processo de simulação foram comparados com os obtidos no artigo do S-MAC [10], de forma a garantir a exatidão da simulação. Após isso, já era possível simular outros protocolos e outros ambientes.

Uma vez obtidos os resultados das simulações, os quais serão apresentados no capítulo seguinte, pôde-se verificar a eficiência do DyTEE, o qual motivou a implementação em módulos reais.

4.2 Escolha da plataforma de implementação

Com o grande número disponível de opções de plataforma de desenvolvimento sem fio no mercado, como Telos [31], IRIS [40], MICAZ [41] e Zigbit [42], torna-se possível escolher aquelas que melhor se adaptam à aplicação desejada. Em aplicações como sistema de automação predial ou rural, os quais tem necessidades específicas, um dos sistemas mais acessíveis e de baixo custo são os módulos que operam na faixa de 2.4GHz, os quais possuem baixo consumo e boa capacidade de transferência de dados.

Dentre essas variedades de plataformas, as opções mais acessíveis, no qual o DyTEE poderia ser desenvolvido, foram o módulo MICAZ (Figura 4.1) e o módulo Zigbit (Figura 4.2).



Figura 4.1: Módulo MICAZ.

Os módulos MICAZ são geralmente trabalhados com o compacto sistema operacional TinyOS [30], o qual é programado com a linguagem nesC [43]. Este módulos são amplamente utilizados na literatura, bem como seu principal sistema operacional, visto que ambos são códigos abertos (em inglês *open source*), permitindo que sejam modificados pelos usuário. As especificações do MICAZ são semelhantes às do Zigbit, pois ambos são voltados a atender às necessidades do padrão IEEE 802.15.4, bem como ambos possuem componentes semelhantes. No entanto, os módulos Zigbit, apresentados na Figura 4.2, são produzidos por outra empresa (Atmel [44]) que proporciona outros dois compactos sistemas operacionais capazes de realizar tarefas parecidas.

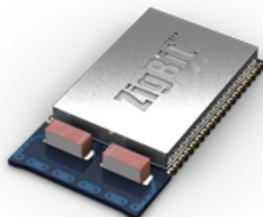


Figura 4.2: Módulo Zigbit (modelo ATZB-24-A2).

O principal sistema operacional utilizado nos módulos Zigbit é chamado de Bitcloud [45]. Este é um sistema fechado, ou seja, não disponibiliza os seus códigos de todas as camadas de comunicação para que possam ser modificados. Dessa forma, não é possível sua utilização para desenvolvimento de protocolos na camada MAC. Entretanto, a segunda opção disponibilizada pela Atmel é o pacote chamado “IEEE 802.15.4 MAC” [46], desenvolvido em linguagem C [47]. Este possui código aberto de todas as camadas (todos os arquivos dos pacote são modificáveis) e tem como principal protocolo o próprio IEEE 802.15.4 [11].

Desta forma, a plataforma Zigbit “IEEE 802.15.4 MAC” foi utilizada para a primeira implementação do DyTEE, pois os módulos são mais compactos e de menor custo, bem como a linguagem desta é convencional. Visando futuros testes em ambientes reais de automação, o fato de possuir menor custo facilita a utilização destes módulos.

4.3 Desenvolvimento em módulo real

Para se utilizar os módulos Zigbit, inicialmente foi necessário o desenvolvimento de placas de de circuito impresso (PCB do inglês *printed circuit board*) capazes de integrar o módulo com saídas e conexões externas, como a serial do computador (USB do inglês *universal serial bus*). A Figura 4.3 (a e b) apresenta o conjunto da placas mais o próprio Zigbit.

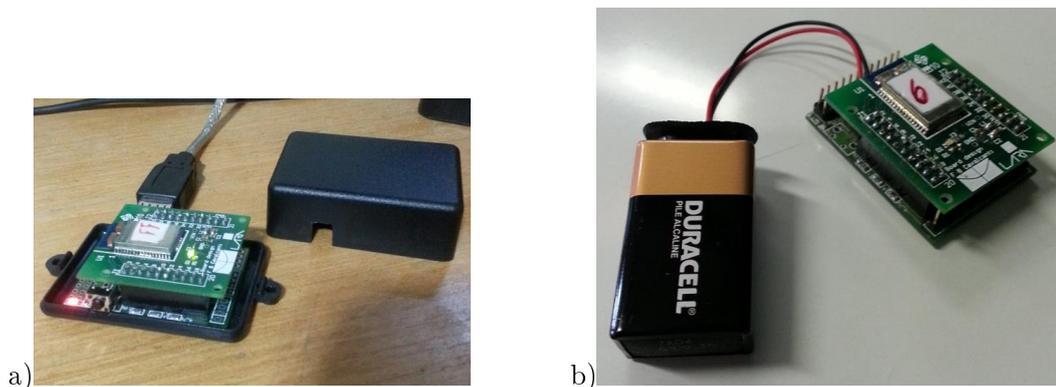


Figura 4.3: Módulo Zigbit integrado com PCB para acesso às saídas: a) com USB; b) Com bateria, sem USB.

As placas que foram desenvolvidas no laboratório LARA (Laboratório de Robótica e Automação), de forma a acoplar o Zigbit, são modelos capazes de oferecer comunicação com o computador (Figura 4.3a) e modelos somente alimentados por bateria, retirando os LEDs e qualquer outra forma de gasto de energia desnecessário (Figura 4.3b). Dessa forma, pode-se trabalhar com um coordenador que transfira as mensagens para um computador e os demais nós somente operando por bateria em outros pontos do ambiente. O esquemático e circuito desta placa estão apresentados nas Figuras I.1 e II.1, respectivamente, dos anexos. É importante ressaltar que a comunicação com o computador, realizada por meio da porta serial (USB), pode ocasionar problemas no sincronismo da execução do protocolo, pois uma vez iniciada a comunicação, o *hardware* do microcontrolador entra em um laço com bloqueio até o fim da transmissão da mensagem (problema também reportado por [48]), ou seja, outras operações do protocolo podem sofrer um atraso e conseqüentemente uma perda de sincronismo. Para resolver este problema, foi criado um armazenador (no inglês *buffer*) de mensagens serial interno no microcontrolador, o qual transmite todas de uma só vez em períodos que não irão afetar o protocolo MAC, ou seja, períodos inativos.

Por meio do acesso USB é feita a gravação do programa que executa o algoritmo do protocolo DyTEE e outros, bem como também os envios dos resultados obtidos dos experimentos, portanto sua necessidade dentro da rede.

Dado que o pacote “IEEE 802.15.4 MAC” disponibilizado pela Atmel possui as camadas já implementadas e necessárias para o módulo (Figura 4.4), o processo de implementação de um novo protocolo foi feito por meio da modificação deste.

Inicialmente foi feita uma verificação de quais funções do protocolo originalmente utilizado ainda poderia ser aproveitada na implementação do DyTEE, como o caso de funções de inicialização do módulo, estrutura de chamada de funções e outras. Entretanto, nesta fase inicial também foram necessárias correções de algumas incompatibilidades do pacote com o Zigbit utilizado.

Com o sistema pronto para trabalhar com as funções mais básicas, como transmissão de dados de um módulo para o outro (diretamente pelo acesso à camada física), foi iniciada a implementação de cada função necessária para o DyTEE, utilizando sempre a mesma estrutura que o pacote “IEEE 802.15.4 MAC” anterior possuía, porém com a nova lógica.

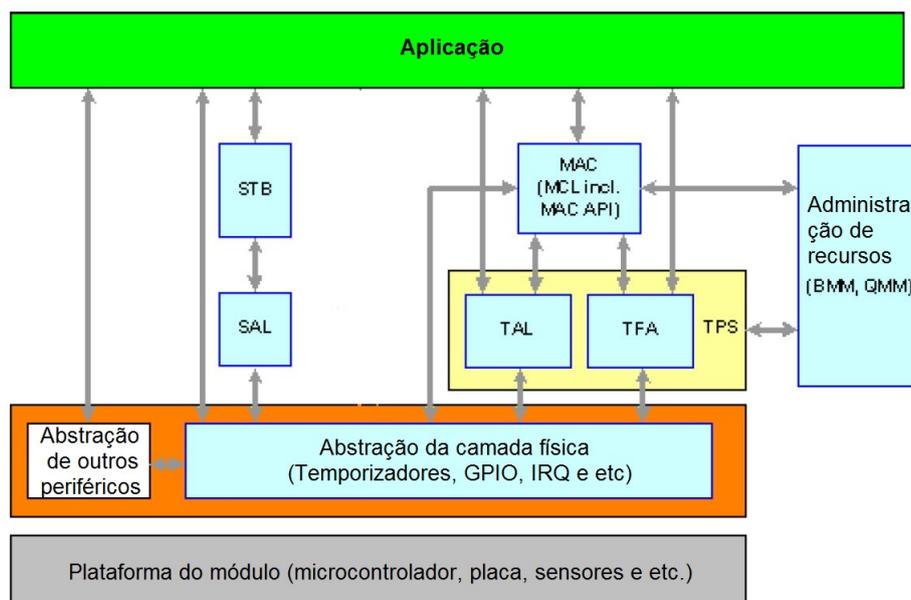


Figura 4.4: Estrutura do pacote aberto IEEE 802.15.4 MAC disponibilizado pela Atmel.

Dessa forma, ao final da implementação, todo o novo pacote continua com a mesma compatibilidade do pacote fornecido pela Atmel, permitindo um intercâmbio de protocolos das camadas de aplicação desejadas.

É importante citar que sempre que uma nova etapa da implementação era desenvolvida, era guardada uma versão do pacote em outra pasta do sistema, de forma a se realizar uma possível recuperação da implementação, caso surgisse algum erro de programação ou lógica no processo, ou seja, entre a implementação bem sucedida de sinais de sincronização e o processo de registro havia uma cópia de segurança deste avanço.

4.4 Obtenção de resultados

O processo de coleta dos dados de simulação em computador é bastante simples, visto que não há dificuldades em acessar os dados da rede. Portanto em simulações, as comparações são facilmente realizadas e colocadas na forma de gráficos. Entretanto, em se tratando de módulos reais, coletar todas as perdas de pacotes, períodos ativos, ociosos e outros, não é tão simples.

4.4.1 Método de contagem de bytes transmitidos e recebidos

Para a coleta de dados e comparação em módulos reais foi então utilizada uma estrutura semelhante à proposta pelos autores do S-MAC-AL [19], onde também foi relatada a dificuldade de coletar dados de tensões e correntes tão baixas por períodos tão rápidos. Esse processo é dado pela determinação indireta dos valores de consumo através dos dados de quantidade de bytes transferidos e recebidos.

Deste modo o consumo em cada estado pode ser inferido em toda a rede por meio do:

- Somatório dos tempos ativos de um nó ($Tempo_{Total\ ativo}$);
- Totais de bytes enviados por um nó ($Bytes_{Totais\ enviados}$);
- Totais de bytes recebidos por um nó ($Bytes_{Totais\ recebidos}$);
- Totais de bytes enviados de carga útil por um nó ($Bytes_{Úteis\ enviados}$);
- Totais de bytes recebidos de carga útil por um nó ($Bytes_{Úteis\ recebidos}$).

O líder do grupo recebe todos os dados dos demais nós da rede, os quais sempre enviam mensagem do mesmo tamanho, no caso 20 bytes úteis mais cabeçalho ($S_{Carga\ útil}$ mais $S_{Cabeçalho}$ formam S_{Pacote}), armazenando somente a última mensagem com os valores atualizados em memória. Ao final da coleta de 50 mensagens da rede, o líder inicia uma transmissão destes dados de forma sequencial para o computador ao qual está conectado.

No computador, estes dados são tratados para interpretar os resultados da rede, nas condições na qual foi testada. Neste caso, por meio do número de bytes úteis recebidos e o tempo de execução do experimento ($Tempo_{Decorrido}$), obtém-se a vazão final da rede, em relação a todos os nós, por

$$Vazão_{Rede} = \frac{\sum Bytes_{Úteis\ recebidos}}{Tempo_{Decorrido}}. \quad (4.1)$$

Também por meio destes dados, pode-se obter a quantidade de pacotes perdidos, pela equação

$$Pacotes_{Perdidos} = \frac{(\sum Bytes_{Úteis\ enviados} - \sum Bytes_{Úteis\ recebidos})}{S_{Carga\ útil}}, \quad (4.2)$$

onde $S_{Carga\ útil}$ é o tamanho padrão de 20 bytes de carga útil (valor selecionado para simular mensagens em sistemas de automação predial e rural).

Retirando-se os bytes dos pacotes perdidos da quantidade total de bytes enviados na rede, tem-se que o excesso de cabeçalho (*overhead*) é dado por

$$Overhead = \frac{(\sum Bytes_{Totais\ enviados} - \sum Bytes_{Úteis\ recebidos} - Pacotes_{Perdidos} \cdot S_{Pacote})}{\sum Bytes_{Totais\ enviados}}, \quad (4.3)$$

onde S_{Pacote} é o tamanho total de um pacote, composto de carga útil e cabeçalho.

Por fim, o consumo total do grupo pode ser obtido pela soma dos consumos individuais de cada estado. Considerando que na configuração e tecnologia 2.4GHz utilizada, demora-se $32\mu s$ por byte transmitido, tem-se que

$$Tempo_{Total\ Tx\ grupo} = 32\mu s * \sum Bytes_{Totais\ enviados} \quad (4.4)$$

e

$$Tempo_{Total Rx grupo} = 32\mu s * \sum Bytes_{Totais recebido}, \quad (4.5)$$

onde Tx e Rx referem-se aos tempos em transmissão e recepção, respectivamente. O que permite inferir que

$$Tempo_{Total ocioso grupo} = \sum Tempo_{Total ativo} - Tempo_{Total Tx grupo} - Tempo_{Total Rx grupo} \quad (4.6)$$

e que

$$Tempo_{Total Sleep grupo} = Nós_{Utilizados} \cdot Tempo_{Decorrido} - \sum Tempo_{Total ativo}, \quad (4.7)$$

onde $Nós_{Utilizados}$ representa o número de nós utilizados durante o experimento, e o tempo em dormência é representado por $Tempo_{Total Sleep grupo}$.

Dados os consumos obtidos nos manuais dos módulos e considerando somente o gasto pelos rádios, tem-se que

$$Consumo_{Total} = Tempo_{Total Tx grupo} \cdot Cons_{Tx} + Tempo_{Total Rx grupo} \cdot Cons_{Rx} + \\ Tempo_{Total ocioso grupo} \cdot Cons_{Ocioso} + Tempo_{Total Sleep grupo} \cdot Cons_{Sleep}, \quad (4.8)$$

onde $Cons_{Tx}$ é o consumo do rádio durante a transmissão, $Cons_{Rx}$ é o consumo do rádio durante a recepção, $Cons_{Ocioso}$ é o consumo do rádio durante períodos ociosos e $Cons_{Sleep}$ é o consumo do rádio durante períodos de dormência.

Portanto, utilizando estes procedimentos, foram obtidos resultados nas simulações e nos dispositivos reais. Nas simulações foram comparados os protocolos S-MAC e DyTEE, enquanto nos experimentos reais foram comparados os protocolos IEEE 802.15.4 e DyTEE, preferencialmente possuindo parâmetro de operação mais próximos possíveis entre si, viabilizando a comparação.

Os pontos de comparação para geração de gráficos foram por meio da variação das taxas de chegada das mensagens. Com isso, no experimento real foram utilizadas:

- A topologia em estrela, com somente taxas de chegadas constantes, variando de 1 até 10 segundos, coletando em cada um dos pontos, no mínimo 10 dados.

Enquanto nas simulações foram realizadas:

- A topologia linear, com taxas constantes, onde pacotes eram criados em períodos fixos, variando de 0 segundo (todas as mensagens eram criadas no primeiro momento e acumuladas

em memória) até 10 segundos;

- A topologia estrela, com taxas constantes, onde pacotes eram criados em períodos fixos, variando de 1 até 10 segundos, reproduzindo os resultados reais.
- A topologia estrela, com taxas aleatórias de média constante para cada simulação (distribuição de Poisson), onde pacotes são criados em momentos aleatórios e em qualquer nó, mas variando de 1 até 10 segundos entre cada simulação.

O critério de parada em todos os casos é dado pela quantidade de mensagens entregues ao último nó (sumidouro), no caso linear, ou ao nó central, no caso estrela. Considerando o trabalho do S-MAC [10], foram utilizadas 20 mensagens de 100 bytes cada, nos procedimentos de simulação com topologia linear, e 50 mensagens de 20 bytes cada, nos procedimentos de simulação e real com topologia estrela.

Dessa forma, após coletadas as informações de todos os pontos (repetidos dez vezes) e considerando que estes possuam uma distribuição T-Student [49], foram obtidas as médias e os intervalos de confiança utilizados para gerar o gráficos das métricas comparadas.

4.4.2 Método de medição em multímetro digital

Visando principalmente uma confirmação dos procedimentos de medições e simulações anteriores, foram realizadas também medidas de corrente elétrica utilizada pelos módulos da rede por meio do multímetro digital modelo 34410A da Agilent [50]. A estrutura básica de medição é dada pela aferição contínua da corrente consumida pelo módulo ao longo do tempo, como pode ser visto na imagem da Figura 4.5.

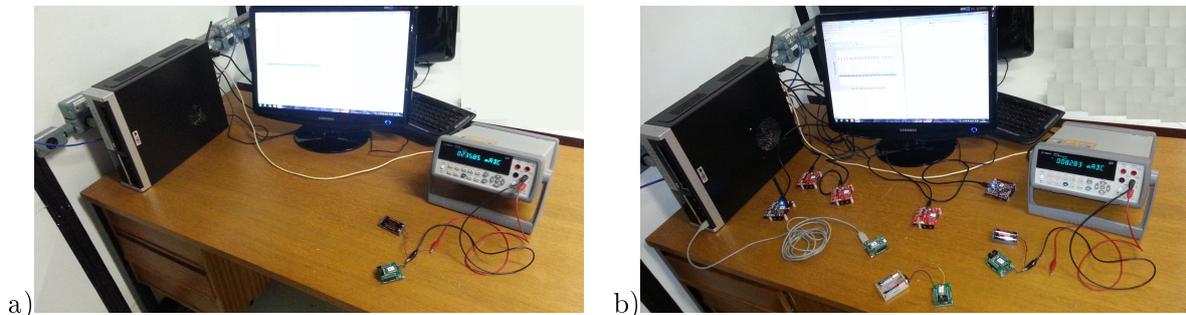


Figura 4.5: Processo de medição de corrente: a) Em um módulo somente; b) Em um módulo dentro da rede de sensores.

Neste processo, realizam-se os mesmos procedimentos, topologia e parâmetros apresentados na Subseção 4.4.1, entretanto considerando agora o consumo energético dado pela aquisição dos dados do multímetro. Este está integrado com o computador, mais especificamente ao software MATLAB [9], o qual recebe os dados e calcula o consumo deste nó, estimando também o consumo de toda a rede com base neste processo.

O processo de cálculo do consumo é dado pela integração dos pontos recebidos pelo multímetro,

dados o tempo de amostragem utilizados (uma medição a cada 0,1ms) e subtraído o consumo médio do próprio controlador do módulo (média dos pontos mais baixos do gráfico). Um exemplo de gráfico obtido por este processo de medição é dado pela Figura 4.6.

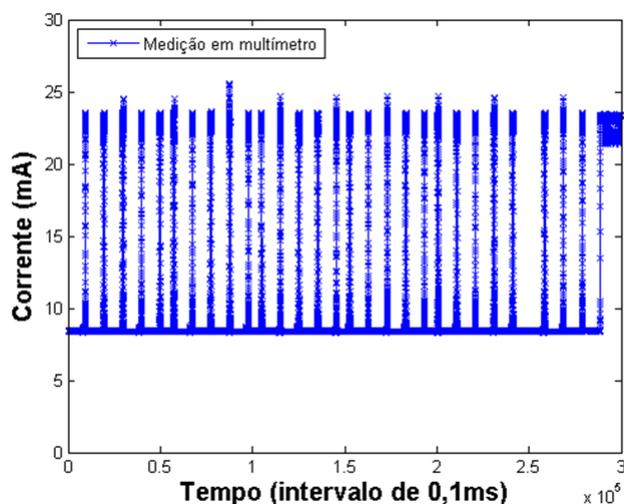


Figura 4.6: Gráfico obtido do processo de medição de corrente de um nó da rede.

Nesta Figura 4.6 pode-se observar que os parâmetros selecionados fazem o protocolo DyTEE realizar períodos ativos a cada um segundo. Alguns problemas já observados neste gráfico deste procedimento serão melhor descritos no capítulo dos resultados.

Ampliando um desses picos com maior consumo entre os demais (como o terceiro da esquerda para a direita) presentes nas medições, pode-se observar como resultado a Figura 4.7.

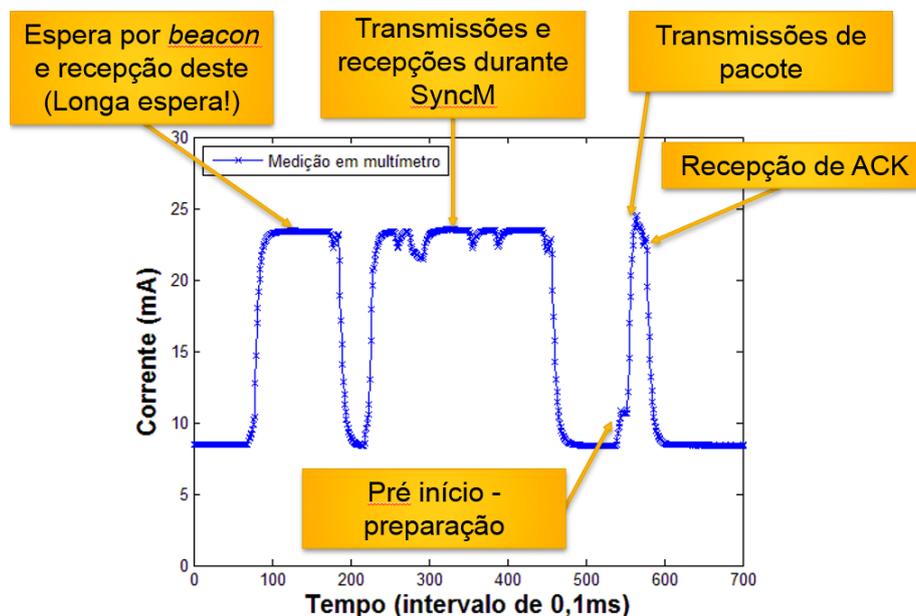


Figura 4.7: Gráfico obtido do processo de medição de corrente de um nó da rede, ampliado em um dos picos.

No gráfico da Figura 4.7 é possível ver que um pico da Figura 4.6 é na verdade um conjunto de variações no consumo de corrente que ocorrem justamente durante os processos de recepção do *beacon*, transmissões e recepções durante o SyncM e troca de pacote com carga útil seguido do ACK (corresponde exatamente ao primeiro período de operação apresentado na Figura 3.1). Nesta implementação foi também utilizado um longo tempo de espera por *beacon*, como pode ser visto no início do gráfico, resultado de uma solução de implementação herdada do pacote anterior “IEEE 802.15.4 MAC” para resolver problemas de perda de sincronismo.

Dessa forma, o processo de medição do consumo é feito facilmente pela integração no tempo destes pontos, vezes a tensão medida no início de cada experimento. É importante lembrar que o nó líder de um grupo consome mais energia do que os demais, porém utilizando dados das simulações é possível estimar este aumento de consumo e então estimar o consumo total de toda a rede. Os experimentos foram repetidos também dez vezes para cada taxa de chegada. Assim, após coletadas as informações e considerando que estes possuam uma distribuição T-Student [49], foram obtidas as medidas e os intervalos de confiança utilizados para gerar o gráficos das métricas comparadas.

4.5 Conclusão do capítulo

A abordagem deste capítulo permitiu determinar os procedimentos e metodologias utilizadas neste trabalho para garantir a correta obtenção dos dados, bem como o sucesso da implementação do novo protocolos nas plataformas escolhidas.

Neste trabalho, a escolha dos parâmetros, métricas e topologias nos experimentos reais foram restritas pela atual disponibilidade de materiais, laboratórios e equipamentos. Dessa forma, a escolha da topologia estrela anteriormente descrita proporciona um ambiente de sensoriamento próximo do existente na automação predial ou rural, criando com poucos módulos uma disputa por recursos semelhante ao encontrado nestes ambientes. Isto também ocorre nas métricas analisadas, as quais podem ser mais facilmente medidas, visto a ausência de uma bancada composta por equipamentos mais adequados, capazes de medir outros parâmetros e até mesmo facilitar a medição da latência entre as trocas de mensagens.

A escolha da plataforma também foi discutida, vista a ampla gama de opções disponíveis hoje. Dessa forma, foi utilizado o módulo Zigbit [42], por ser simples, compacto e possuir maior acessibilidade, principalmente quanto ao seu custo. Entretanto, nesta escolha foram observados certas complicações da plataforma, bem como do pacote do sistema operacional “IEEE 802.15.4 MAC” utilizado.

O equipamento disponível e utilizado neste trabalho, sem contar os próprios módulos sem fios e computadores, foi o multímetro digital Agilent 34410A [50]. Neste foi possível coletar a corrente consumida por alguns módulos e estimar o consumo do protocolo para o resto da rede. É importante citar ainda que pela falta de uma bancada mais completa, não foi possível coletar o consumo de todos os módulos sem fio ao mesmo tempo durante um experimento. Nas simulações não existem estes problemas, mas pela falta de possibilidade de comparar com os módulos reais, as simulações possuíram as mesmas métricas adicionada da medição de latência entre as trocas de

mensagens.

Contudo, após a determinação da topologia e dos parâmetros que atenderiam à aplicação de automação predial e rural, estes procedimentos permitiram a coleta de dados que comparam os resultados do novo protocolo com os demais estudados neste trabalho.

Capítulo 5

Resultados e discussões

Este capítulo apresenta os resultados obtidos tanto nas simulações, quanto nos experimentos realizados, os parâmetros utilizados para estes e os gráficos comparativos em cada situação.

Como comentado anteriormente, os primeiros resultados obtidos foram com simulações comparativas entre o S-MAC e o DyTEE, utilizando os parâmetros semelhantes aos utilizados na primeira implementação do S-MAC [19]. Após este, foram realizadas as implementações e obtidos resultados experimentais, de forma a adaptar a simulação com parâmetros dos módulos atuais e então comparar a nova simulação do DyTEE e os resultados experimentais. Por fim, o protocolo S-MAC foi adaptado para os novos parâmetros dos módulos utilizados e então novamente comparado com o DyTEE.

Esses resultados serão apresentados a seguir, utilizando as métricas latência (nas simulações), vazão, excesso de cabeçalho, consumo (totais e parciais por estados) e uma métrica combinada que é a composição da vazão com o consumo de energia.

5.1 Simulações com parâmetros S-MAC

Considerando o trabalho realizado no desenvolvimento do S-MAC [10], o qual foi replicado e, juntamente com o DyTEE, implementado em simulação sob as mesmas condições expostas pelo autor, tem-se então que os parâmetros utilizados (disponíveis em outro trabalho do S-MAC [19]) são apresentados na Tabela 5.1. É importante observar que o consumo no estado ocioso não foi disponibilizado e por tanto considerado o mesmo valor do estado de recepção.

Nesta simulação foi utilizada a topologia linear (Seção 2.2.1), de forma a reproduzir os resultados e o processo utilizado pelo autor do S-MAC. Portanto ambos protocolos foram simulados segundo essa estrutura, na qual as mensagens são geradas só no módulo número 1 (Figura 2.10) e variam-se somente as taxas de chegada das mensagens. O critério de parada da simulação é dado pelo número de mensagens que chegam ao nó final, sendo que neste caso são esperadas 20 mensagens ao final do processo, cada uma com 100 bytes de carga útil.

Tabela 5.1: Parâmetros usados para simular os protocolos na topologia linear.

Parâmetro		Valor	Unidade
Cabeçalhos		8	Byte
Pacotes de controle		10	Byte
Carga útil		100	Byte
Taxa de transmissão		20	Kbps
Ciclo de trabalho		10	%
Tempo máximo ativo		115	ms
Número de nós (não contando o destino)		10	nós
Consumo de energia:	Recepção	14.4	mW
	Transmissão	36	
	Dormindo	0,015	

A Figura 5.1 apresenta a quantidade de energia consumida por todos os nós da rede em ambos protocolos.

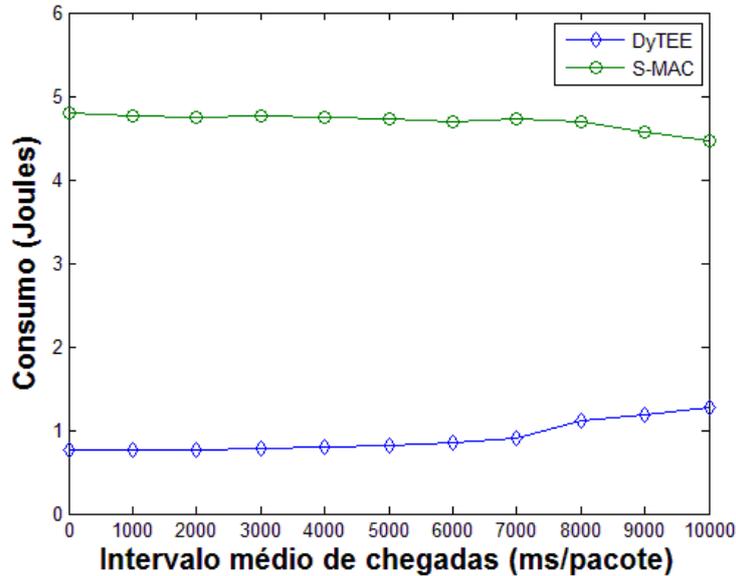


Figura 5.1: Total de energia consumida por todos os nós para enviar 20 pacotes de 100 bytes cada, da fonte para o nó de destino.

Como pode ser visto, o DyTEE possui um melhor desempenho que o S-MAC com aproximadamente 1/5 do consumo de energia. Isso se dá pelo fato que o DyTEE, na maior parte do tempo possui menos nós ativos do que o S-MAC, bem como possui menores tempos de atividade para atender a mesma tarefa, ao contrário do S-MAC no qual todos os nós do grupo estão ligados durante os períodos ativos, os quais podem durar mais do que no DyTEE.

Outra importante diferença está na forma de utilização dos períodos ativos. O S-MAC utiliza o mesmo procedimento em todos os períodos ativos: vagas para sincronizar o grupo e vagas para trocar pacotes de controle (*slots* com disputa pelo canal para o envio de RTS/CTS), seguido da

transmissão efetiva de um pacote com carga útil. Assim o S-MAC somente efetiva uma transmissão com carga útil a cada período ativo, após as trocas dos pacotes de controle, sendo estes passíveis de colisão. Enquanto no DyTEE, existe um período ativo para organização do grupo, com troca de pacotes de controle, seguido de outros períodos ativos com transmissões de pacotes com carga útil, ou seja, para cada período ativo de controle, vários outros períodos ativos já são organizados para haver somente transmissões úteis, isso sem que haja colisões em nenhuma etapa.

Isso pode ser visto no gráfico de ativação dos nós em ambos protocolos da Figura 5.2.

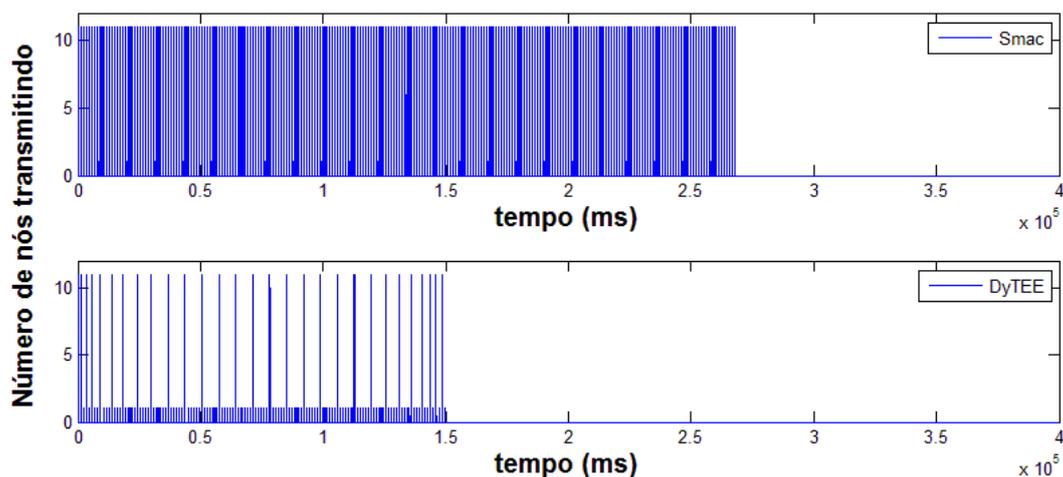


Figura 5.2: Série temporal da quantidade de nós ligados e recebendo por tempo em ambos protocolos.

Ampliando-se um trecho da Figura 5.2, pode-se observar melhor o comportamento comparativo entre os dois protocolos, dado pela Figura 5.3.

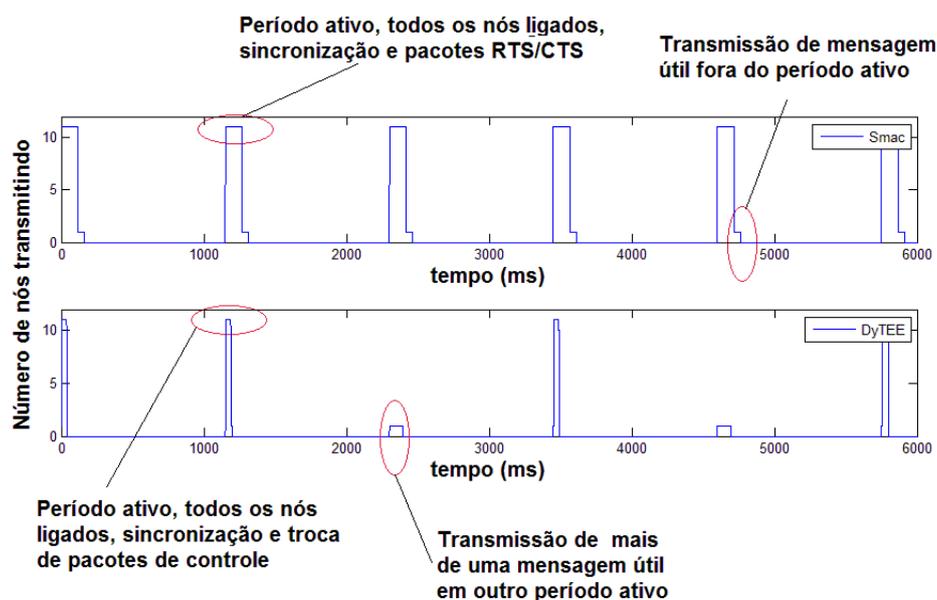


Figura 5.3: Série temporal da quantidade de nós ligados e recebendo por tempo em ambos protocolos.

Como apresentado nas Figuras 5.2 e 5.3, o DyTEE possui essa melhor economia de energia baseado tanto no tempo total necessário para transportar o mesmo número de mensagens na rede, quanto na quantidade de nós ligados por tempo. Isso pode ser visto pela área azul em ambos os gráficos, a qual representa os nós ligados e aponta que o S-MAC possui mais nós ativos por tempo do que o DyTEE. Ainda observando a Figura 5.3, pode-se ver que quando ocorre um período ativo (praticamente todos os nós ligados) nos dois protocolos, o DyTEE passa menos tempo neste momento do que o S-MAC, justificando novamente o alto rendimento observado.

Em relação aos gastos com transmissões, a Figura 5.4 apresenta a quantidade de nós ligados em transmissão durante este experimento linear. Além de também se observar a maior duração do experimento utilizando o S-MAC em relação ao DyTEE, pode-se ver que no DyTEE não ocorrem colisões (onde se vê mais de um nó transmitindo ao mesmo tempo), o que já ocorre no gráfico do S-MAC. Isso proporciona um aumento do gasto energético deste protocolo.

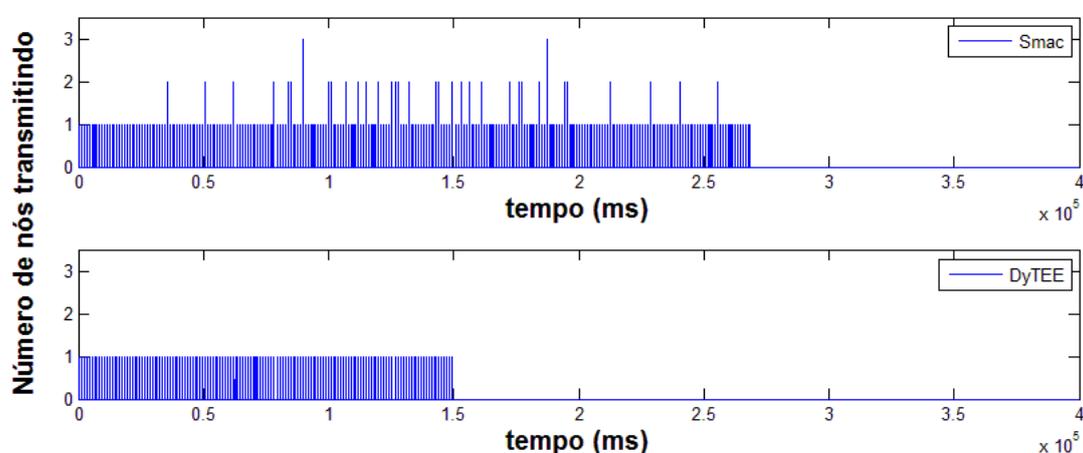


Figura 5.4: Série temporal da quantidade de nós ligados e transmitindo por tempo em ambos protocolos.

Entretanto, devido à existência de um líder de grupo, existe um maior gasto proporcional deste coordenador em relação aos demais. Isso é tratado pela alternância de líderes, como explicado no Capítulo 3. Em uma aplicação de automação predial ou automação da agricultura, o consumo do líder pode não ser relevante, pois este pode estar conectado à rede elétrica no supervísório. Neste caso, o parâmetro que inicia a alternância de líderes (consumo de uma porcentagem da carga da bateria, por exemplo) nunca é ativado, mantendo a rede estável da mesma forma. Contudo, por meio dessa alternância, o DyTEE também pode atender demais aplicações com líderes não conectados aos supervísórios. Assim, a Figura 5.5 apresenta o maior consumo percebido pelo líder do grupo.

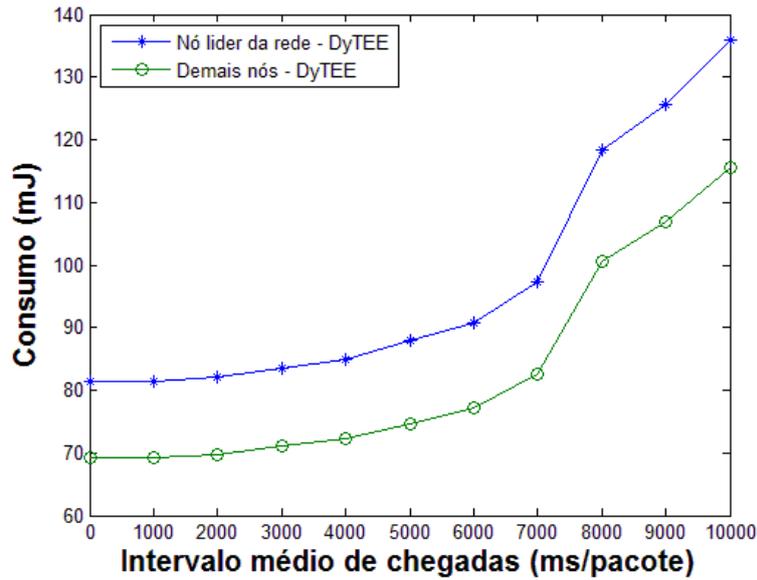


Figura 5.5: Consumo médio do nó líder comparado à média dos demais.

Com base na Figura 5.5, mesmo se o nó líder possuir um consumo mais elevado devido ao gasto para sincronização do grupo, as simulações mostraram que o consumo de um líder está em torno de 10,8% a mais do que os outros nós. Como mencionado antes, a eficiência energética é o alvo principal do protocolo DyTEE. Visto que o D-MAC [32] trabalha com cerca da metade da energia gasta pelo S-MAC, o DyTEE atingiu um consumo cerca de $\frac{2}{9}$ do S-MAC. Da mesma forma, essa maior eficiência também é observada nas Figuras 5.1, 5.6 e 5.7 que apresentam os resultados de consumo total da rede, das latências e excesso de cabeçalho (*overhead*) obtidos dessas simulações, respectivamente.

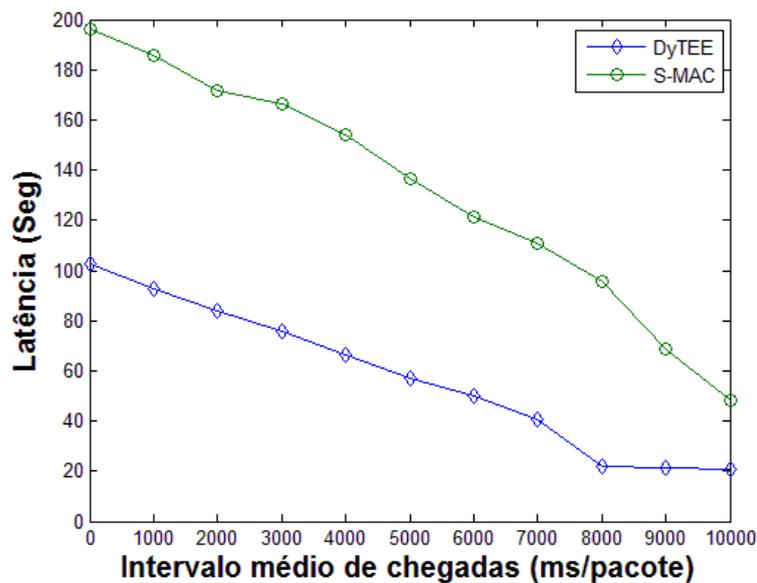


Figura 5.6: Média de latência de pacotes para o envio do nó fonte ao nó de destino.

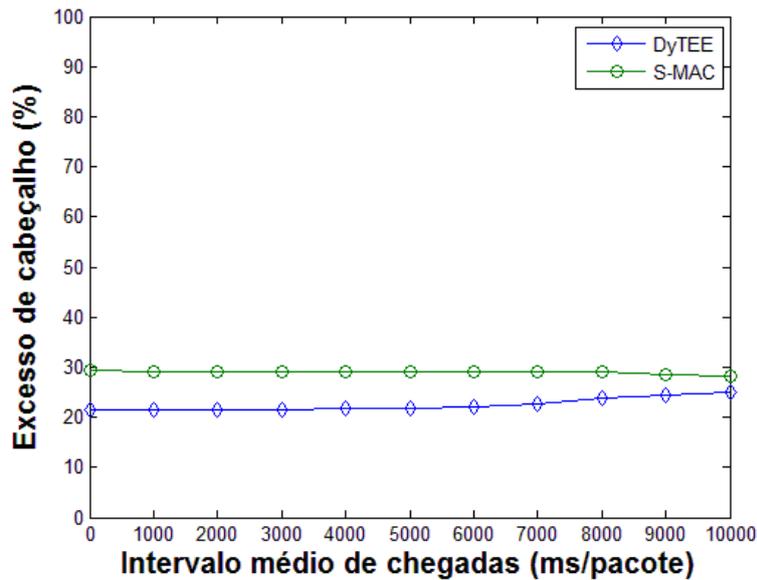


Figura 5.7: Excesso de cabeçalho calculado no processo.

O ganho em desempenho também é observado para a taxa de transferência (vazão), que foi avaliada em relação a toda a rede, ou seja, refere-se ao número médio de bytes por segundos de todas as transmissões dos nós somados. Este resultado é mostrado na Figura 5.8.

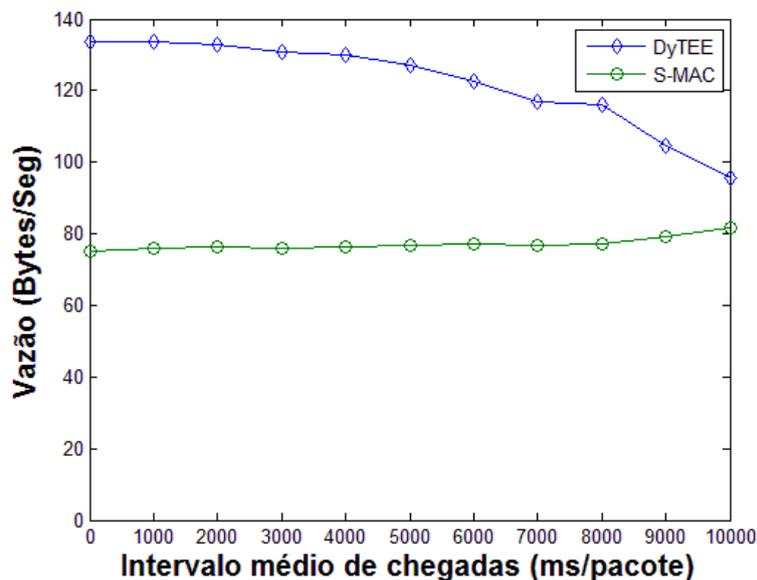


Figura 5.8: Vazão ao longo de toda a rede.

A razão destes melhores resultados é que enquanto no S-MAC se utiliza um período ativo inteiro (com todos os nós ligados) somente para a organização e transferência de uma mensagem, no DyTEE este período ativo, contendo o evento SyncM, já organiza e possibilita a transferência de várias mensagens que serão transmitidas nos períodos seguintes. Dessa forma, o ganho de eficiência

é proporcional ao número de nós que desejam transmitir dentro da rede. Portanto, quanto menor for o intervalo de chegada de mensagens, maior será a quantidade de nós querendo transmitir e conseqüentemente, maior será o ganho do DyTEE em relação ao S-MAC. Ainda que na pior das hipóteses, onde nenhum nó quer transmitir, o DyTEE mostrará consumo menor do que o S-MAC, porque menos nós estão ligados ao mesmo tempo no grupo durante todo o período ativo, que também é menor (Figura 5.3).

5.2 Simulações com parâmetros atualizados

Utilizando os parâmetros da implementação realizadas nos módulos atuais (Zigbit), inclusive as taxas de perda de pacotes (apresentadas na seção seguinte), foram feitas novas simulações atualizando as características do S-MAC para atender ao novo sistema desejado, o qual possui parâmetros que atendem também aos requerimentos do IEEE 802.15.4. Sendo assim, nesta simulação foram assumidos valores como maiores taxas de transferência, menores tempos de vagas para envio de controle de pacote e outras mudanças que permitiriam implementar o S-MAC nos módulos atualmente utilizados. Dessa forma os valores implementados em simulação são diferentes entre os dois protocolos, mas ainda muito próximos, e são apresentados pela Tabela 5.2.

Tabela 5.2: Parâmetros usados para simular os protocolos na topologia estrela.

Parâmetro	Valor S-MAC	Valor DyTEE	Unidade
Cabeçalhos	5	8	Byte
Pacotes de controle	10	8	Byte
Carga útil	20	20	Byte
Taxa de transmissão	250	250	Kbps
Ciclo de trabalho	4.5	5.6	%
Tempo máximo de ativo	45	56	ms
Número de nós (não contando o destino)	8	8	nós
Consumo de energia:	Recepção	51.15	mW
	Transmissão	54.45	
	Ocioso	25.74	
	Dormindo	0,000066	

Segundo os dados utilizados na Tabela 5.2, a pequena diferença nos valores de ciclo de trabalho e tempo máximo ativo são relativos ao fato de que o S-MAC utiliza um tempo do período de dormência para transmitir, sendo assim, para melhor comparar ambos protocolos foi deixada essa diferença, o qual junto com o ciclo trabalho selecionado gera um período total da rede de 1 segundo.

Nesta nova simulação foram consideradas também os gastos de energia em momentos ociosos, os quais não estavam inclusos nos parâmetros fornecidos pelo autor (Tabela 5.1). Dessa forma o consumo de ambas as redes foi obtido, segundo as taxas de chegadas constantes e aleatórias, resultando nos gráficos das Figuras a seguir.

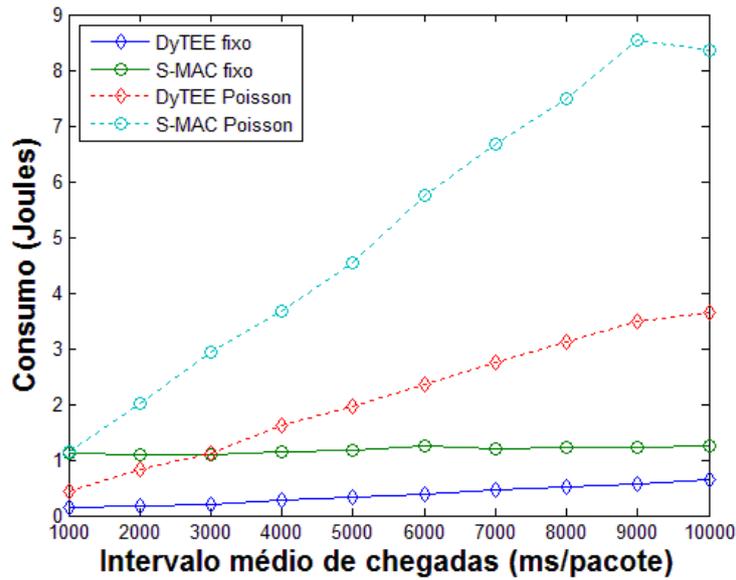


Figura 5.9: Total de energia consumida em ambos protocolos segundo diferentes tipos de chegada de mensagens.

Como pode ser visto, com os parâmetros atualizados, o DyTEE também apresenta melhores resultados do que o S-MAC, tanto na situação de mensagens chegando ao mesmo tempo para todos os nós (taxa constante, representadas pelas linhas contínuas), quanto no caso em que as mensagens chegam aleatoriamente em qualquer nó da rede (segundo Poisson, representadas pelas linhas pontilhadas).

A vazão é apresentada na Figura 5.10.

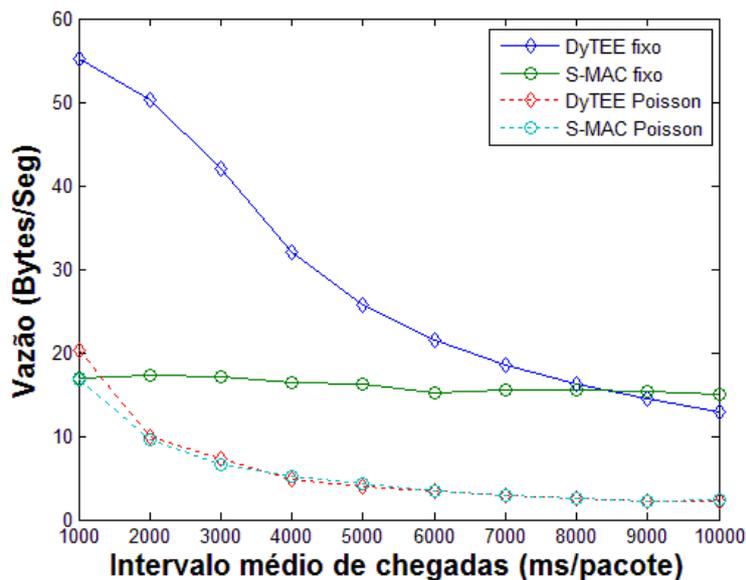


Figura 5.10: Vazão de ambos protocolos segundo diferentes tipos de chegada de mensagens.

No caso da Figura 5.10, pode-se observar que o DyTEE tende a ter um melhor desempenho quando a rede está mais congestionada. Isso reflete que em taxas de chegadas mais baixas, menor congestionamento, o protocolo tende a possuir piores resultados do que o S-MAC, isso pois o excesso de ocorrências de períodos SyncM sem a existência de nós querendo transmitir causa um gasto desnecessário tanto de energia quanto de outros parâmetros nesta rede. Isso também é verificado nas Figuras 5.11 e 5.12.

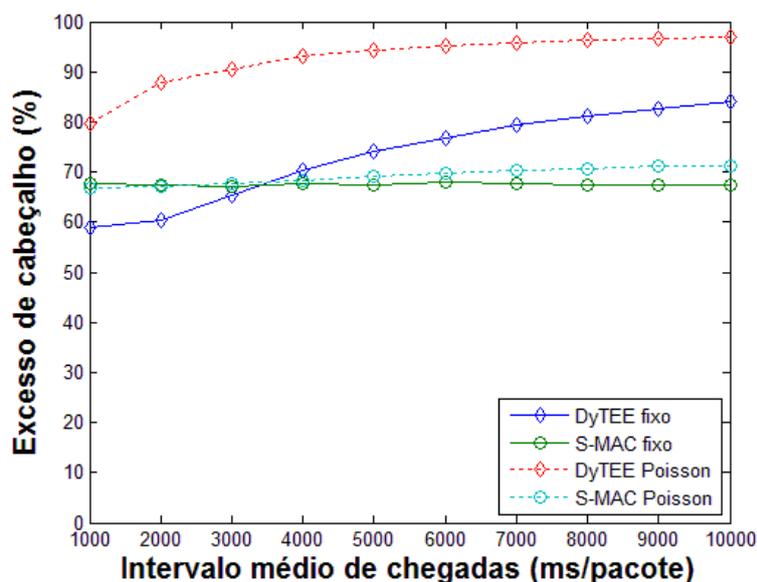


Figura 5.11: Excesso de cabeçalho de ambos protocolos segundo diferentes tipos de chegada de mensagens.

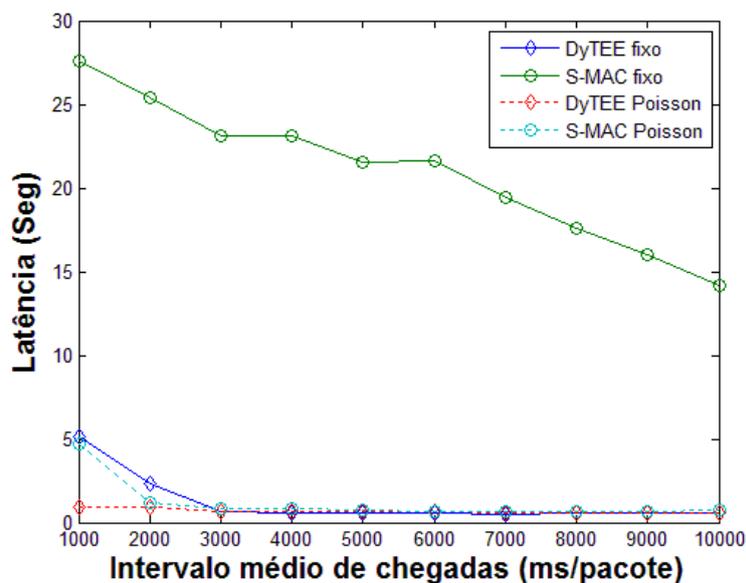


Figura 5.12: Latência de ambos protocolos segundo diferentes tipos de chegada de mensagens.

Segundo os parâmetros adotados na Tabela 5.2, o protocolo DyTEE operou com períodos de 1 segundo, bem como o S-MAC. Dessa forma os momentos de sincronização (SyncM) ocupam em média quase todo um período ativo e portanto foram necessários pelo menos dois períodos inteiros para a transmissão de todos os dados de todos os nós. Neste sentido, observa-se nos gráficos que o protocolo DyTEE opera melhor após o ponto de 2 segundo de média de chegadas de mensagens, ou seja, o ponto onde o DyTEE se torna capaz de melhor atender o sistema.

É importante lembrar que estes parâmetros foram escolhidos para testar vários pontos de operação deste protocolo e são alteráveis, ou seja, no caso de uma aplicação específica, é possível configurar este protocolo de forma que se tenha um melhor resultado do que o apresentado neste experimento. Entretanto, visto que a chegada de mensagens pela distribuição de Poisson evita a geração simultânea de vários pacotes, reduzindo a disputa pelo canal, pode-se observar pelas Figuras 5.10 e 5.11 que o DyTEE tende a possuir piores resultados em sistemas menos congestionados, diferente do S-MAC que opera com vazão semelhante ao DyTEE neste caso.

Portanto, utilizando o tempo de duração de cada experimento e a quantidade de energia total que um nó gasta durante o experimento, obtém-se o gráfico da Figura 5.13.

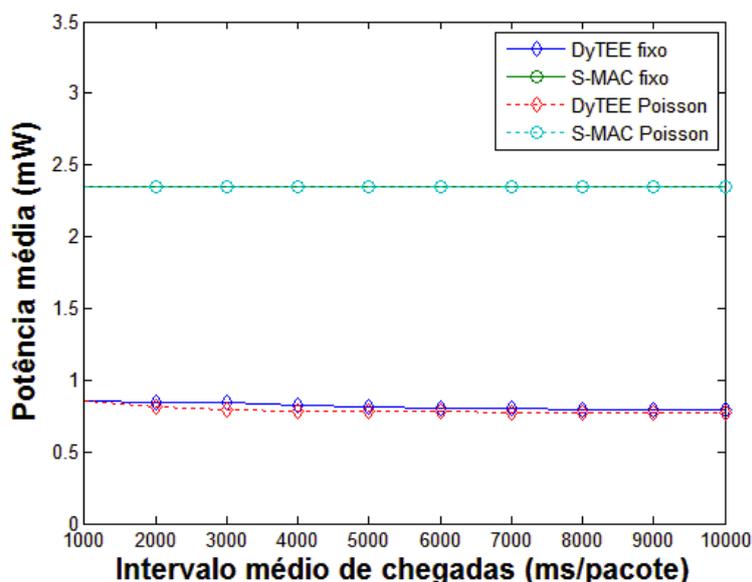


Figura 5.13: Consumo médio de um nó em ambos protocolos segundo diferentes tipos de chegada de mensagens.

Por fim, foi utilizada a métrica combinada, envolvendo vazão, energia total consumida do grupo (Figura 5.9) e latência. O valor numérico não é muito representativo, mas sim a distância entre os pontos de cada protocolo. Portanto a Figura 5.14 apresenta essa nova comparação, de forma a melhor avaliar os resultados entre ambos protocolos.

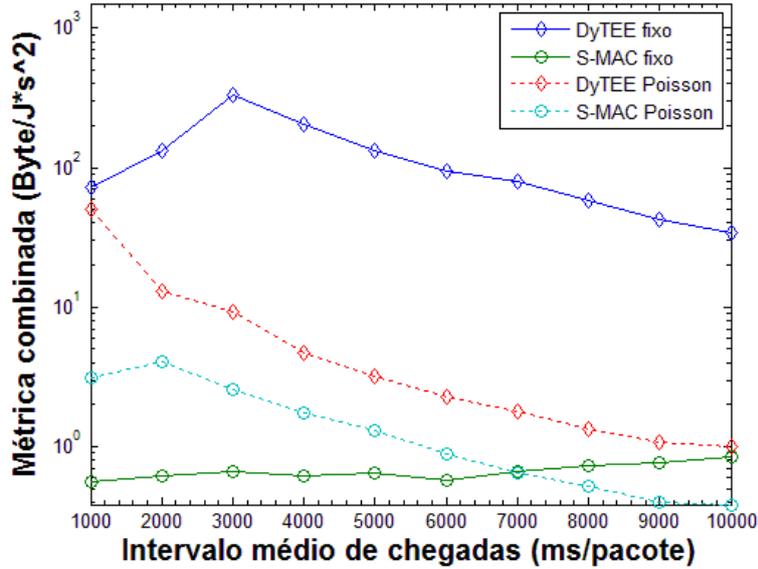


Figura 5.14: Gráfico da combinação de três métricas relacionando ambos protocolos.

Essa métrica combinada tripla é dada por

$$Métrica_{Tripla} = \frac{Vazão}{Latência_{Média} \cdot Consumo_{Total}}, \quad (5.1)$$

onde os pontos de vazão, latência e consumo são dados pelos pontos das Figuras 5.10, 5.12 e 5.9, respectivamente.

Como pode ser vista nesta nova métrica, o desempenho do DyTEE é muito superior, principalmente após o ponto de 2 segundos entre as taxas de chegada, situação no qual os parâmetros escolhidos melhor atendem ao sistema.

5.3 Resultados do DyTEE implementado em módulos reais

Após implementado o DyTEE e realizado experimentos em módulos reais, os resultados foram comparados com as simulações, de forma a averiguar a proximidade da simulação atualizada e utilizada neste trabalho. Neste âmbito, algumas métricas vistas anteriormente foram comparadas entre a versão real e simulada, e são apresentadas nas Figuras seguintes.

5.3.1 Método de contagem de bytes transmitidos e recebidos

Considerando o primeiro método de análise de consumo em experimentos reais mencionado na metodologia (Subseção 4.4.1), a vazão de ambos os experimentos é dado pela Figura 5.15, onde a simulação assumiu os mesmos valores de parâmetro que a implementação.

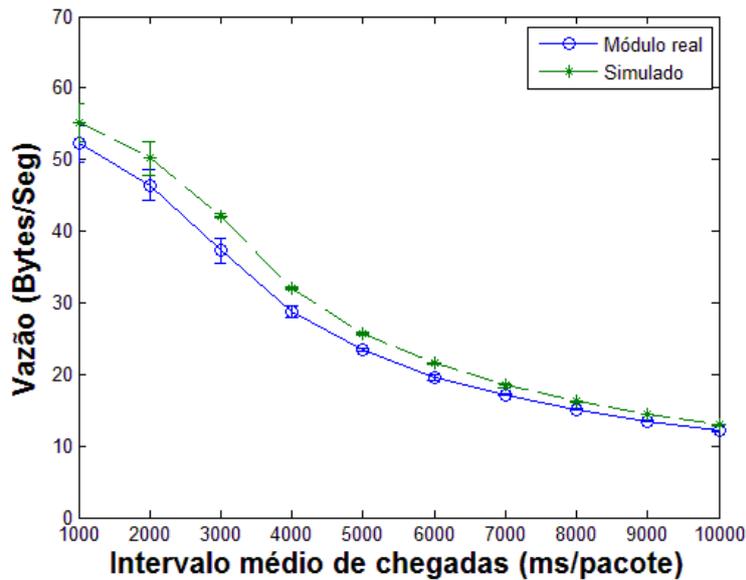


Figura 5.15: Vazão na rede simulada e real.

Observa-se então que o ambiente simulado se comporta muito próximo do protocolo implementado nos módulos reais, com pequenas diferenças, pois nas operações reais ocorrem falhas e dessincronismos que não são esperados pela simulação. Se consideradas as dimensões utilizadas nos gráficos, pode-se ver que os dois sistemas, real e simulado, operaram muito semelhantemente também com relação ao consumo, o qual é apresentado na Figura 5.16 e separadamente em seus quatro estados na Figura 5.17.

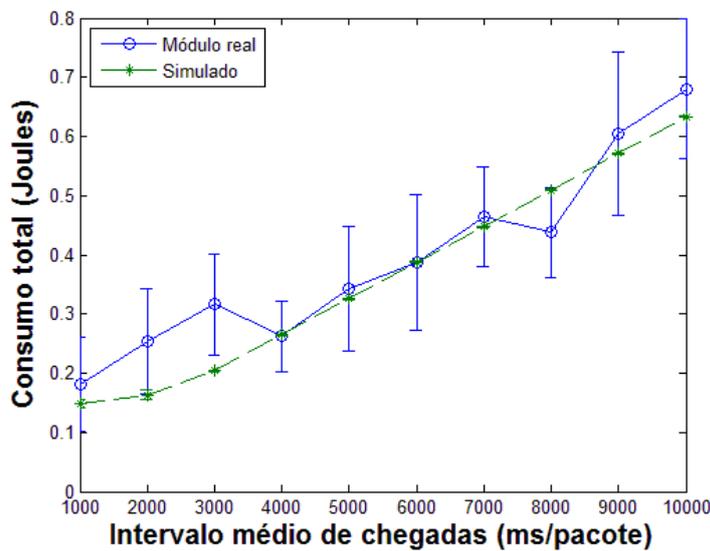


Figura 5.16: Consumo total de energia da rede simulado e real.

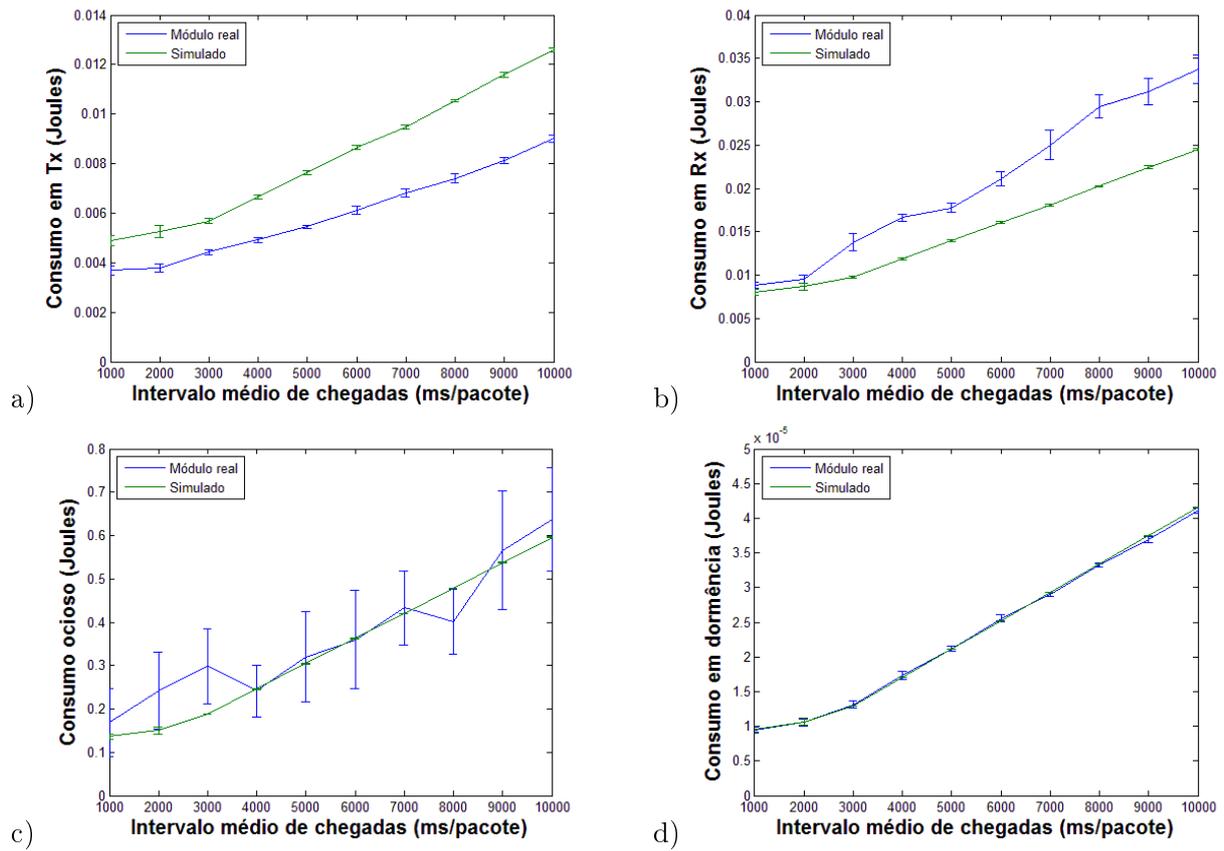


Figura 5.17: Consumo total de energia da rede em seus quatro estados: a) Transmitindo; b) Recebendo; c) Ocioso e d) Em dormência.

Em relação às demais métricas, as Figuras 5.18 e 5.19 apresentam essas comparações, onde novamente se observa a semelhança dos resultados simulados com os resultados reais do DyTEE.

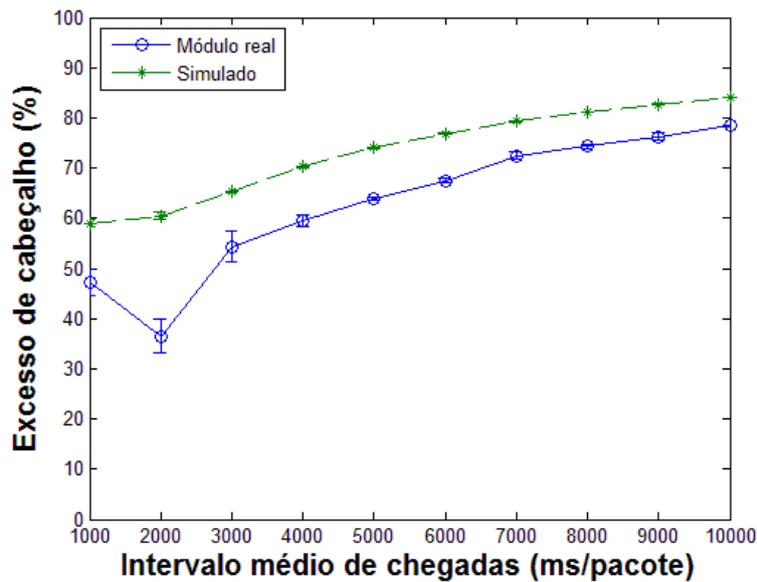


Figura 5.18: Excesso de cabeçalhos entre os experimentos reais e simulados.

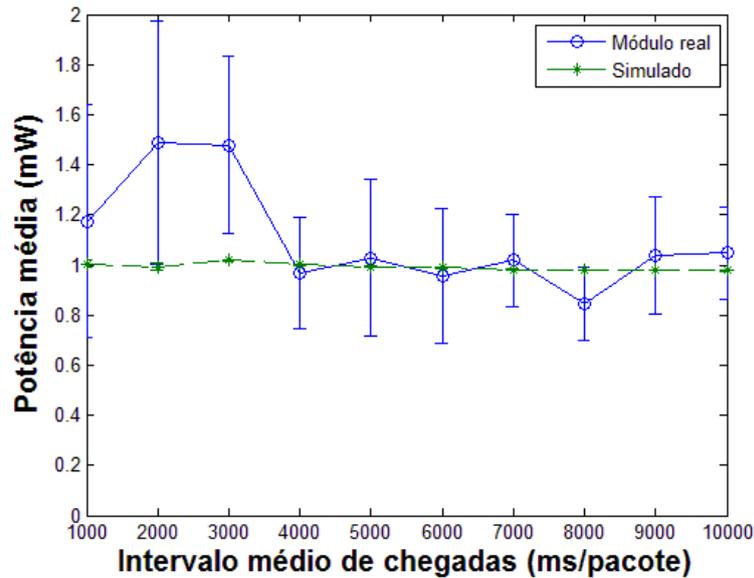


Figura 5.19: Potência média consumida por um módulo operando o protocolo DyTEE nos procedimentos reais e simulados.

Como pode ser visto, os gráficos de consumo possuíram valores muito semelhantes, confirmando que o processo de medição real utilizado está corretamente relacionado à simulação realizada.

5.3.2 Método de medição em multímetro digital

Verificando os resultados obtidos na Subseção 5.3.1, agora por meio da medição por multímetro digital, foi observado que durante o processo de execução do protocolo DyTEE na plataforma utilizada, existiram certos problemas de perda de sincronismo. Isso pode ser visto na Figura 5.20(a), pelas regiões de corrente elétrica fora do padrão esperado, ou seja, somente picos de transmissões e recepções como o obtido na Figura 5.20(b).

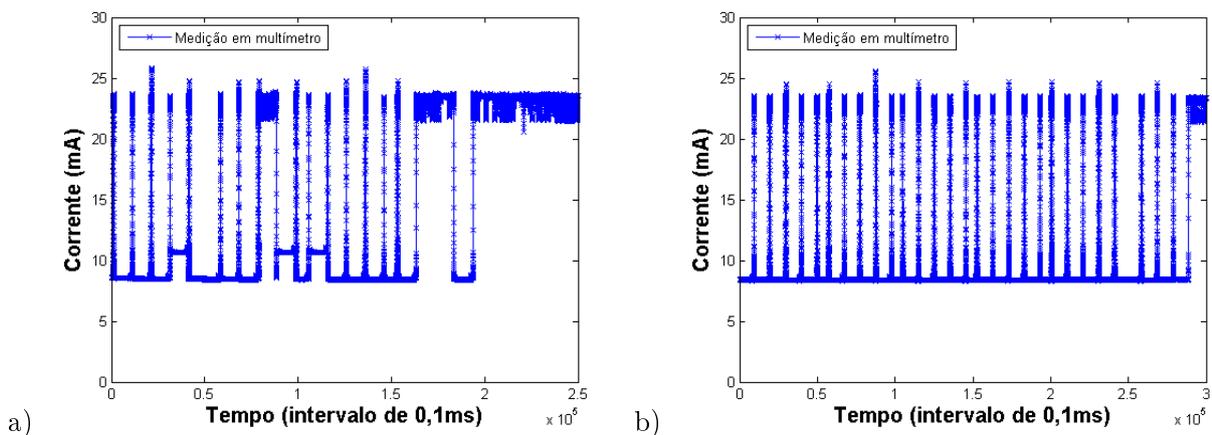


Figura 5.20: Comparação medição da corrente elétrica do protocolo DyTEE: a) Nos experimentos com falhas de implementação/plataforma; b) No padrão desejado.

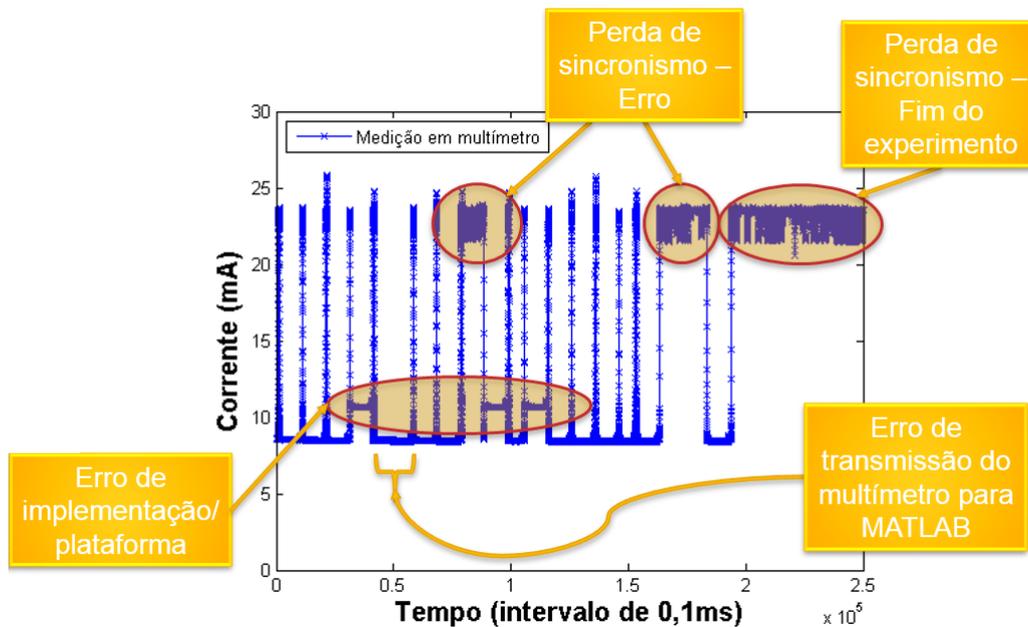


Figura 5.21: Gráfico da medição de corrente elétrica do protocolo DyTEE de um módulo na rede apresentando erros de implementação/plataforma.

Destacando-se as regiões de falhas de implementação/plataforma (Figura 5.21), pode ser observado que algumas destas apresentam padrões de consumo semelhantes aos de quando o experimento chega ao fim, o que comprova que naquele momento ocorreu uma perda de sincronismo do módulo com o líder, entretanto retornando ao modo operacional logo no próximo período. Isso demonstra a capacidade do protocolo DyTEE em retomar sua funcionalidade em casos de perda de sincronismo ou falhas (sistema robusto).

Existem ainda outras regiões com erros de medição causados pela comunicação entre o multímetro e o computador, os quais geram atrasos na medição e deslocamento dos pontos do gráfico. Isso é relacionado ao problema de tamanho de memória disponível no multímetro utilizado, que no caso possui o máximo de 50.000 pontos a serem guardados. Dessa forma, como foi utilizada a resolução de um ponto cada 0,1ms, tem-se que a cada 5 segundos de experimento, ocorre um envio de dados do multímetro para o computador. Neste processo de envio, alguns pontos do experimento não estão sendo medidos, logo, ao final da coleta, surgem os espaçamentos ou aproximações entre os picos observados na Figura 5.21.

Contudo, como foram realizados vários experimentos para cada taxa de chegada de mensagem, na média os pontos obtidos foram muito próximos da simulação e do método de contagem de bytes, se considerados os intervalos de confiança, como pode ser visto no gráfico de consumo energético da Figura 5.22.

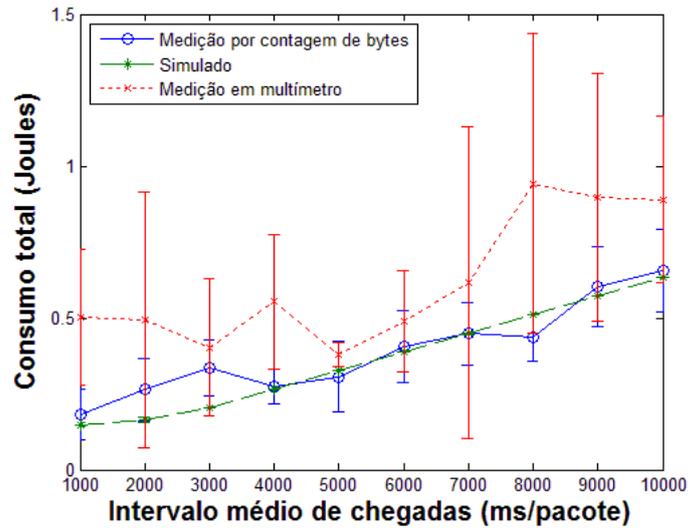


Figura 5.22: Gráfico de consumo total do protocolo DyTEE, comparado aos demais métodos de avaliação.

No gráfico de consumo, pode-se ver também que as tendências dos pontos em todos os métodos seguem da mesma forma, entretanto com valores do método de multímetro levemente mais altos, como esperado se considerado que nos demais processos de medição, os erros de implementação e plataforma são menos sentidos do que no processo de medição direta por multímetro.

O resultado da avaliação da potência média também pode ser visto no gráfico da Figura 5.23.

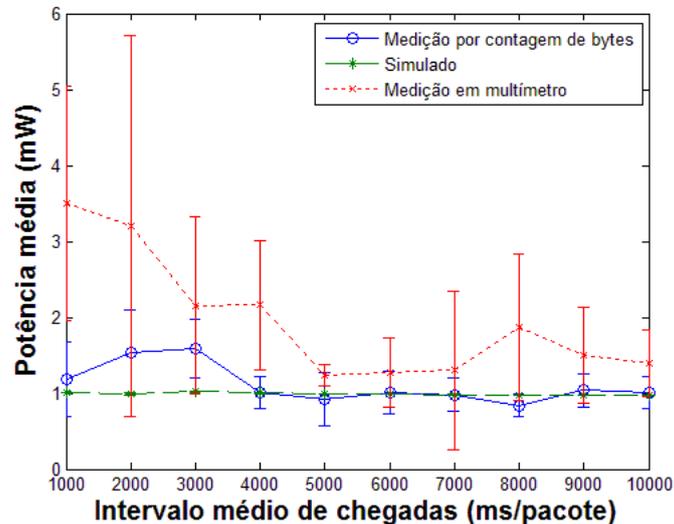


Figura 5.23: Gráfico da potência média consumida por um nó executando o protocolo DyTEE, comparado aos demais métodos de avaliação.

Como os erros de implementação/plataforma discutidos anteriormente influenciam muito nos resultados da avaliação, foi questionado também como resultariam estes gráficos de consumo e potência, caso estes erros não estivessem presentes (um resultado ideal do experimento). Para isto,

foi feita uma avaliação adicional do protocolo DyTEE com pontos alterados manualmente, ou seja, onde ocorriam trechos com erros, foram substituídos por trechos do mesmo experimento que não possuem erros, o que resulta em gráficos de corrente como apresentados anteriormente na Figura 4.6.

Assim, corrigidos os erros de implementação e plataforma, o consumo total da rede operando o DyTEE é dado pela Figura 5.24.

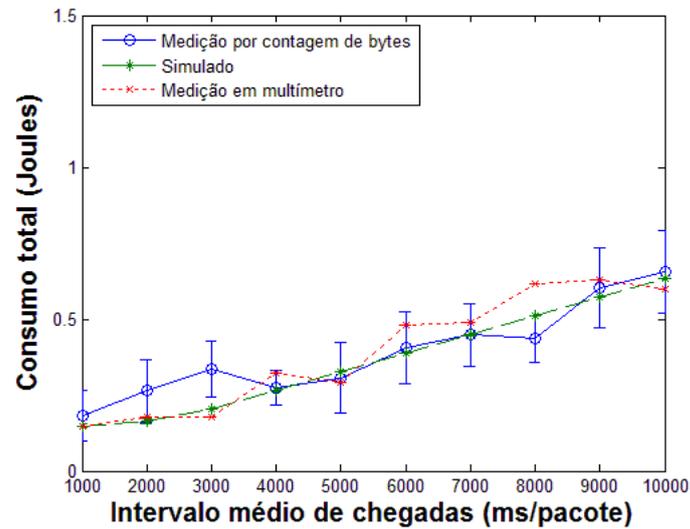


Figura 5.24: Gráfico de consumo total do protocolo DyTEE, sem erros de implementação/plataforma, comparado aos demais métodos de avaliação.

Pode-se ver então que, após substituídos os erros observados na Figura 5.21, os pontos de todos os métodos de avaliação ficam ainda mais próximos, o que também é presenciado na Figura 5.25.

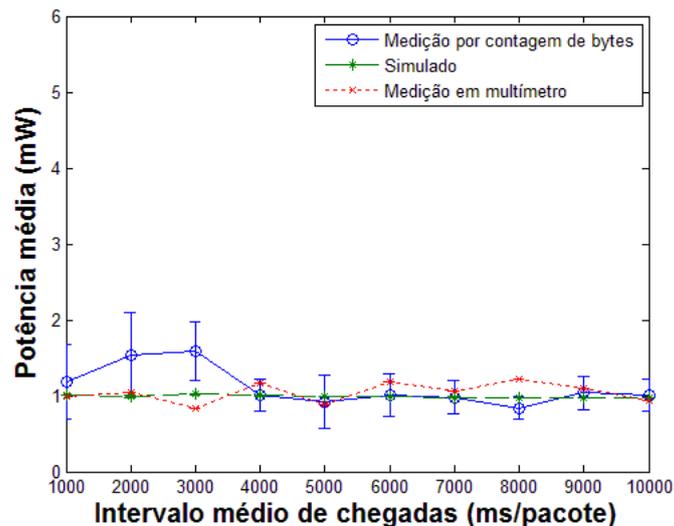


Figura 5.25: Gráfico de consumo total do protocolo DyTEE, sem erros de implementação/plataforma, comparado aos demais métodos de avaliação.

Dessa forma, conclui-se que os resultados nos três métodos são válidos e seguem as mesmas tendências.

5.4 Comparação do DyTEE e o IEEE 802.15.4 em módulos reais

A escolha da plataforma Zigbit utilizada, como comentado no capítulo anterior, permitiu uma rápida implementação do DyTEE em módulos reais, por meio da modificação do pacote fornecido pela Atmel (IEEE 802.15.4 MAC) [46]. Neste pacote há como principal protocolo o próprio IEEE 802.15.4, o qual possui vários modos de configuração e operação.

Entretanto, nos próprios exemplos disponíveis neste pacote, encontrou-se certas incompatibilidades entre este e o *hardware* Zigbit utilizado (Figura ??). Esse problema foi resolvido após estudos mais aprofundados do pacote e então pode-se utilizar o protocolo IEEE 802.15.4 para a comparação com o DyTEE.

Contudo, certas configurações e operações do IEEE 802.15.4 ainda não funcionaram corretamente na plataforma escolhida. Assim, o período mínimo factível de duração dos *superframes* (estrutura de organização em *slots* realizado pelo IEEE 802.15.4 para que cada nó e coordenador transmita suas mensagens e requisições) foi de ordem 5, ou seja, aproximadamente 500 ms. Isso prejudica a comparação entre os dois protocolos, pois não permite que o IEEE 802.15.4 opere em parâmetros que seriam mais próximos aos utilizados pelo DyTEE.

Os parâmetros que funcionaram e permitiram comparar com o IEEE 802.15.4 foram:

- Ordem 5 de *superframe*, o que resulta aproximadamente 500 ms de tempo ativo máximo do grupo;
- Ordem 6 de intervalos de *beacons* (sinais de sincronização enviados no início de todo o ciclo de trabalho, soma dos períodos ativos e não ativos), o que equivale aproximadamente a 1 segundo de período total;
- Foram utilizados também 8 módulos, sendo um deles o coordenador;
- Rede centralizada e coordenada, com envio de *beacons*;
- Tamanho da identidade igual a 2 bytes (endereços curtos).

Outros problemas de sincronismo, atrasos e requisições com duração variável trouxeram uma grande dificuldade durante a implementação do protocolo DyTEE nestes módulos, visto a utilização do pacote “IEEE 802.15.4 MAC”. O protocolo IEEE 802.15.4 presente originalmente no pacote possui também certos problemas, como desconexão de nós, registro em grupo falhando (para ordem de *superframe* inferior a 5), mensagens da serial enviadas com caracteres adicionais entre outros. Considerando estes problemas já existentes nesta plataforma utilizada, a implementação do DyTEE pode ser considerada bem sucedida, visto que atingiu melhores resultados do que o protocolo IEEE 802.15.4 sob as mesmas circunstâncias, bem como realizou as tarefas mais rapidamente.

Nesta comparação entre protocolos foi utilizado somente o método de contagem de bytes, visto a facilidade e a validação obtida na comparação entre os métodos no protocolo DyTEE apresentado na Subseção 5.3.2.

Dessa forma o consumo total da rede em ambos protocolos pode ser visto na Figura 5.26.

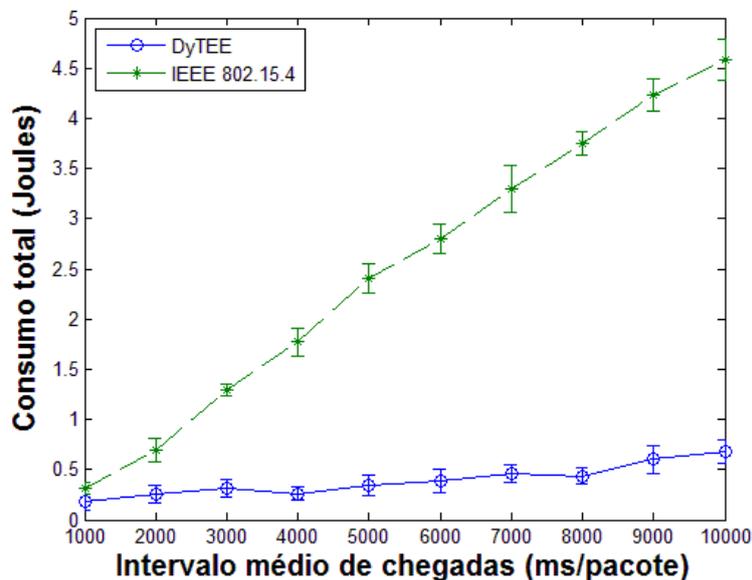


Figura 5.26: Total de energia consumida por todos os nós para enviar 50 pacotes de 20 bytes cada.

Devido ao tempo ativo no protocolo IEEE 802.15.4, o aumento de consumo por aumento de taxa de chegadas é bem maior do que no DyTEE, bem como é mais linear, pois no DyTEE esse consumo ocorre dinamicamente de acordo com o número de mensagens existentes na rede.

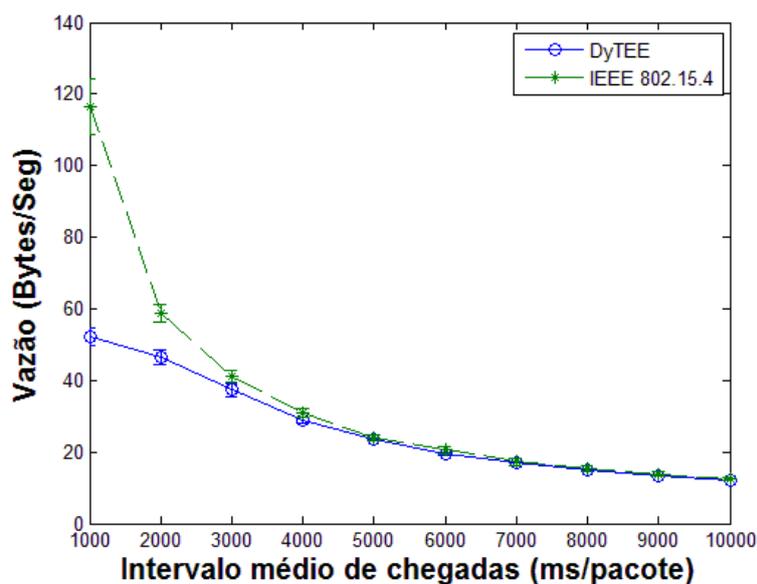


Figura 5.27: Vazão obtida em ambos protocolos em módulos reais.

Já no gráfico da vazão (Figura 5.27), obteve-se uma curva semelhante à observada nas simulações, onde o DyTEE não possui os parâmetros¹ ideais para operar em intervalos de chegadas inferiores a 2 segundos por pacote. Visto a pouca ocorrência de erros de transmissão e sincronismos, no ponto de 2 segundos de taxas de entregas o DyTEE está sob máxima operação. Isso pode ser visto pelo fato de que a partir deste pontos, ambos protocolos atendem ao sistema com vazões semelhantes, as quais são somente limitadas pela taxa de chegadas.

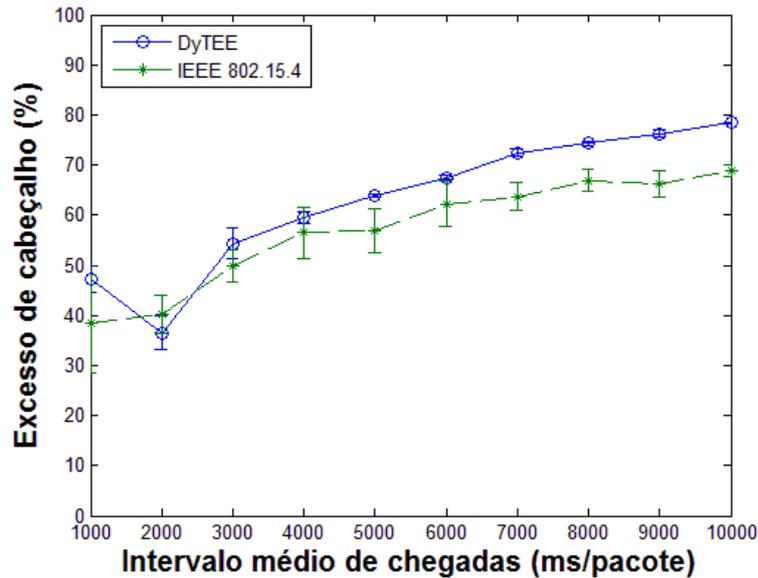


Figura 5.28: Excesso de cabeçalho de ambos protocolos em módulos reais.

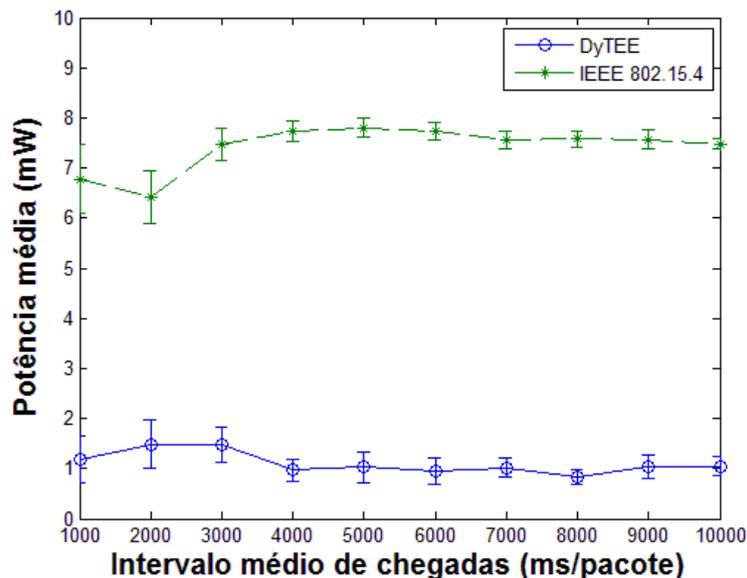


Figura 5.29: Potência média consumida por um módulo real em ambos protocolos.

¹Os parâmetros do protocolo DyTEE escolhidos neste trabalho geram a necessidade de no mínimo dois períodos completos (o que neste caso dura 2 segundos) para o envio de todas as mensagens recebidas pelos nós.

O excesso de cabeçalho, visto na Figura 5.28, apresenta que o DyTEE ainda possui maiores valores de desperdício com cabeçalho, entretanto essa troca entre maiores cabeçalhos permite uma organização mais eficiente e mais rápida, o que justifica sua melhor economia de energia e potência média de consumo apresentado no gráfico da Figura 5.29.

No caso dos experimentos reais, não foram coletadas as latências, pois segundo a plataforma utilizada não havia um método rápido de implementação desta métrica. Isso pois, para a avaliação deste parâmetro, seria necessário a modificação do protocolo DyTEE para inserção de valores de tempo nos quais as mensagens seriam criadas, ou acréscimo do tamanho dos pacotes (maiores do que 20 bytes) para inserção de marcas de tempo. Essa modificação acarretaria em um aumento de cabeçalho e conseqüentemente um pior desempenho. Outro método de aferir a latência entre os nós seria a utilização de ferramentas externas, mas que devido às circunstâncias no qual este protocolo foi desenvolvido, não foi possível ser realizada ainda.

Portanto, utilizando uma ideia semelhante à equação (5.1), no caso dos experimentos reais foi proposta uma métrica combinada mais simples, composta de somente dois termos, sendo então

$$Métrica_{Dupla} = \frac{Vazão}{Consumo_{Total}}, \quad (5.2)$$

onde o resultado é apresentado na Figura 5.30.

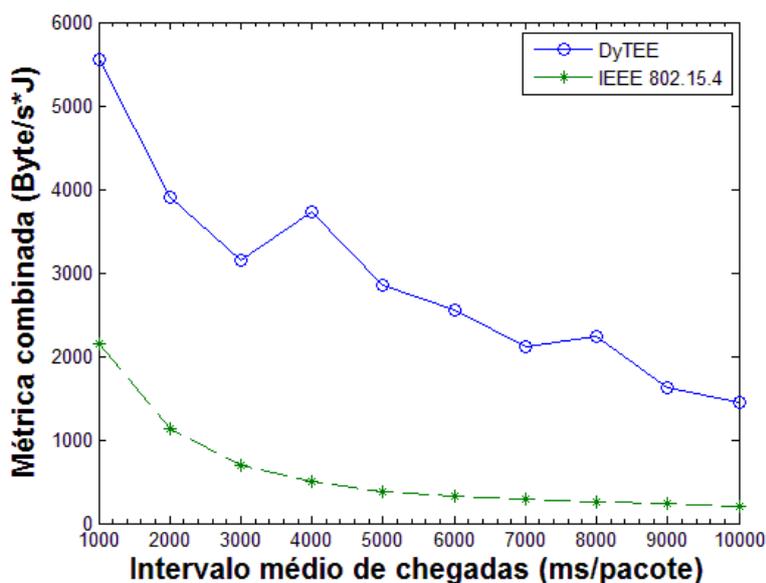


Figura 5.30: Métrica combinada dupla para ambos protocolos em módulos reais.

Novamente, os pontos deste gráfico não possuem um valor muito significativo, mas sim a sua magnitude em relação ao ponto do outro protocolo. Sendo assim, observa-se que o DyTEE possui um melhor desempenho do que o IEEE 802.15.4 durante todos os pontos de taxa de chegadas, mesmo nos casos onde o protocolo não foi projetado para operar de forma ótima (taxas de chegadas igual a 1 segundo).

Por fim, o tamanho total do programa o qual foi inserido nos módulos reais de forma a se obter o protocolo DyTEE funcionando foi de 34.582 bytes, sendo 6.638 bytes de dados. Esses valores são respectivos a todo o programa utilizado para executar o protocolo DyTEE, mais a camada de aplicação e o processo de coleta de dados para aferir métricas.

A mesma análise foi feita para o IEEE 802.15.4 já presente no pacote anteriormente, onde o programa utilizado para executar a mesma tarefa e compará-lo com o DyTEE possui, com a funcionalidade de coordenador, um tamanho total de 38.870 bytes (sendo 3.114 bytes de dados) e, com a funcionalidade de folha, um tamanho total de 38.436 (sendo 1.980 bytes de dados).

A implementação do DyTEE foi genérica em relação à funcionalidade dos nós, enquanto o programa que executa o IEEE 802.15.4 foi dividido entre coordenador e folhas, o que indica que uma implementação genérica do IEEE 802.15.4 ocuparia ainda mais espaço em memória. Portanto o DyTEE foi implementado ocupando um espaço de memória inferior ao IEEE 802.15.4, porém com um pouco mais de variáveis de dados, o qual pode ter sido ocasionado pela modificação do pacote ao invés da recriação deste.

5.5 Tempo de vida do módulo real

Baseado nos dados apresentados anteriormente e nos parâmetros determinados para a execução destes experimentos, pode-se estimar o tempo de vida (duração máxima de funcionamento) de um módulo operando com o protocolo DyTEE e com o protocolo IEEE 802.15.4.

Considerando uma aplicação de medição de temperatura de uma sala, onde a taxa de coleta de dados seja de uma medida a cada cinco segundos (ponto equivalente a 5000ms de taxa de chegada dos experimentos) e os módulos configurados exatamente como utilizados nos experimentos anteriores (períodos de *beacon* de 1 segundo e 8 módulos conectados), tem-se então que o protocolo DyTEE irá consumir 1,028mW, enquanto o IEEE 802.15.4 consumirá 7,799mW.

Visto que uma bateria genérica capaz de fornecer 2000mAh a 3,1V, o que poderia representar um par de pilhas do tipo AA, possui uma carga em Joules dada por

$$Carga_{Bateria} = (2A * 3,1V) * 1h * 60min * 60seg = 22320J, \quad (5.3)$$

e que a plataforma utilizada neste trabalho (Figura 4.3) consome aproximadamente 8,2mA, com o *hardware* do rádio em dormência (*sleep*), medido por instrumentos [50]. Pode-se inferir que o consumo total deste módulo integrado a um sensor de temperatura SHT71 [51], que segundo *datasheet* pode consumir em média $90\mu W$, é dado por

$$Consumo_{Módulo} = (8,2mA * 3,1V) + 0,09 = 25,51mW. \quad (5.4)$$

Assim, o tempo de vida deste módulo operando o DyTEE é dado por

$$Tempo_{DyTEE} = \frac{22320J}{(25,51 + 1,028)mW} \cong 841058,11seg \cong 9,73dias. \quad (5.5)$$

Entretanto, devido ao maior consumo do IEEE 802.15.4, o tempo de vida deste é dado por

$$Tempo_{DyTEE} = \frac{22320J}{(25,51 + 7,799)mW} \cong 670089,17seg \cong 7,75dias. \quad (5.6)$$

Pode-se perceber que a diferença no consumo dos protocolos causou um ganho de aproximadamente 25,6% no tempo de vida destes módulos, ou seja, 2 dias a mais de funcionamento. É importante citar que este tempo de vida ainda poderia ser maior, caso os parâmetro fossem melhor ajustados para esta aplicação.

5.6 Conclusão do capítulo

Neste trabalho foram realizados vários experimentos contendo diferentes parâmetros e topologias. Entre as simulações realizadas, foram comparados os protocolos S-MAC [10] e o novo protocolo DyTEE, tanto na topologia linear com parâmetros informados pelo mesmo autor em [19], quanto na topologia estrela com parâmetros da implementação em módulos reais aqui trabalhados. Nos experimentos em módulos reais, foram comparados os protocolos IEEE 802.15.4 [11] e o novo protocolo DyTEE.

Dentre os três métodos de avaliação utilizados (simulação, contagem de bytes e medição por multímetro), foi observado que todos possuem pontos muito próximos, principalmente quando os erros de implementação/plataforma foram corrigidos idealmente. Isso permite inferir que os métodos são válidos, apesar de suas leves variações, bem como que o método de contagem de bytes possui menor sensibilidade aos erros de implementação/plataforma.

Os resultados obtidos nestes experimentos permitiram então observar que o DyTEE possui melhor desempenho do que ambos S-MAC e IEEE 802.15.4. Assim, como visto na estimativa de tempo de vida de um módulo sensor de temperatura, o protocolo DyTEE permite até 25,6% a mais de duração do que o protocolo IEEE 802.15.4, bem como possui um consumo energético de aproximadamente 1/5 do S-MAC.

Assim, o DyTEE superou estes demais protocolos em diversas métricas, inclusive na métrica combinada, a qual relaciona vazão, consumo de energia e, quando considerado, latência. Isso se deve principalmente ao fato de os pacotes de controle trocados permitirem a organização prévia de vários *slots* de transmissão que ocorrem em períodos posteriores, assim aumentando a eficiência do sistema sem a utilização de pacotes RTS ou CTS.

Capítulo 6

Conclusões e Perspectivas

6.1 Conclusões finais

Este trabalho propõe um novo protocolo, denominado *Dynamic Timed Energy Efficient* (DyTEE), combinando características do TDMA, CSMA, S-MAC e outros protocolos para remover a colisão de dados e, dessa forma, apresentar baixo consumo de energia comparado com os protocolos S-MAC e IEEE 802.15.4.

Das simulações, tanto nas duas topologias (linear e estrela) quanto nos diferentes parâmetros testados, observou-se que o DyTEE superou o desempenho do S-MAC em latência, excesso de cabeçalho e vazão. Foi perceptível também a necessidade de se ajustar corretamente os parâmetros da rede, de forma mais específica para cada aplicação, para que o DyTEE opere com melhor desempenho em cada caso, pois em sistemas muito relaxados (com poucas mensagens sendo transmitidas entre os nós) há uma tendência de se possuir maiores gastos com cabeçalho e períodos ativos desnecessários.

Dos experimentos em dispositivos reais, comprovou-se a eficiência do DyTEE comparado ao IEEE 802.15.4, entretanto com eficiência em menor proporção do que em relação ao S-MAC, isso pois o IEEE 802.15.4 também possui uma grande variedade de configurações que permitem melhor ajuste para cada aplicação. Vale observar que nos experimentos reais foram também avaliados os parâmetros de consumo energético e potência média por meio de medição em multímetro digital, o qual comprovou a eficiência do método inicialmente utilizado de contagem de bytes.

Contudo, observando sempre o ponto de operação da rede para o qual os parâmetros foram projetados (intervalos de chegadas de 2 segundos por pacote), o DyTEE apresenta sempre o maior ganho (métrica combinada) tanto em relação ao S-MAC, quanto em relação ao IEEE 802.15.4.

Dado o foco do protocolo DyTEE em aplicações com características específicas (seção 1.3 do capítulo 1), este trabalho somente realizou testes considerando estes ambientes. Outras aplicações com foco em baixas latências ou altas vazões não foram testadas. Entretanto, ajustando os parâmetros do DyTEE, os resultados atuais apontam que este protocolo também atenderia às

necessidades de outras aplicações.

6.2 Perspectivas

O potencial do DyTEE ainda não foi completamente explorado, portanto trabalhos futuros podem aprimorar este protocolo. As principais sugestões que ainda necessitam ser estudadas são:

- Implementar o DyTEE em outras plataformas confiáveis e com melhor suporte de outras pesquisas, como no sistema TinyOS [30] dos motes MICAZ [41], ou no sistema ERIKA [52] em outros motes semelhantes ao MICAZ, como realizado em [53];
- Estudar maior redução do uso de pacotes de cabeçalho, o qual ainda possuiu grande impacto visto que o DyTEE foi implementado pelas modificações de um pacote (IEEE 802.15.4 MAC) e não pela recriação deste;
- Comparar o DyTEE em diferentes topologias com outros parâmetros e pacotes com maiores cargas de informação;
- Testar este protocolo em uma aplicação realmente operante, como automação de uma sala ou irrigação de um campo;
- Desenvolver uma integração entre camadas para associar a organização do acesso ao canal (camada MAC) com o roteamento das mensagens, ou seja projeto entre camadas (do inglês *crosslayer design*), possibilitando maiores economias da rede utilizando as características da camada MAC projetada em conjunto com a camada de roteamento, a qual também possui estudos de protocolos de roteamento com foco em eficiência em energia [33];
- Aprimorar a camada MAC DyTEE por meio da junção deste protocolo com outro mais simples, de forma a atender aplicações que possuam grande variabilidade em suas taxas de geração de mensagem, de forma que quando a rede estiver mais relaxada, um protocolo mais simples entre em atuação, economizando ainda mais energia.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ADERE, K.; MURTHY, G. R. Solving the hidden and exposed terminal problems using directional-antenna based MAC protocol for wireless sensor networks. In: IEEE. *Wireless And Optical Communications Networks (WOCN), 2010 Seventh International Conference On*. 2010. p. 1–5.
- [2] HANDZISKI, V.; POLASTRE, J.; HAUER, J.-H.; SHARP, C. Flexible hardware abstraction of the TI MSP430 microcontroller in tinyOS. In: ACM. *Proceedings of the 2nd international conference on Embedded networked sensor systems*. 2004. p. 277–278.
- [3] DEMIRKOL, I.; ERSOY, C.; ALAGOZ, F. MAC protocols for wireless sensor networks: a survey. *Communications Magazine, IEEE*, IEEE, v. 44, n. 4, p. 115–121, 2006.
- [4] PRAYATI, A.; ANTONOPOULOS, C.; STOYANOVA, T.; KOULAMAS, C.; PAPADOPOULOS, G. A modeling approach on the TelosB WSN platform power consumption. *Journal of Systems and Software*, v. 83, n. 8, p. 1355 – 1363, 2010. Performance Evaluation and Optimization of Ubiquitous Computing and Networked Systems.
- [5] GUIMARÃES, V. G.; BAUCHSPIESS, A. AUTOMAÇÃO E MONITORAMENTO REMOTO DE SISTEMAS DE IRRIGAÇÃO VISANDO AGRICULTURA FAMILIAR. In: *Proc. of XIX Congresso Brasileiro de Automática (CBA)*. Campina Grande, PB, Brazil, 2012.
- [6] DOUDOU, M.; DJENOURI, D.; BADACHE, N. Survey on latency issues of asynchronous MAC protocols in delay-sensitive wireless sensor networks. *Communications Surveys & Tutorials*, IEEE, v. 15, n. 2, p. 528–550, 2013.
- [7] GUIMARÃES, V. G.; BAUCHSPIESS, A.; MORAES, R. M. D. AN ENERGY EFFICIENT MAC PROTOCOL FOR WIRELESS SENSOR NETWORKS IN AUTOMATION ENVIRONMENTS. In: *Proc. of XI Simpósio Brasileiro de Automação Inteligente (SBAI)*. Fortaleza, CE, Brazil, 2013.
- [8] FALCONER, D. D.; ADACHI, F.; GUDMUNDSON, B. Time division multiple access methods for wireless personal communications. *Communications Magazine, IEEE*, v. 33, n. 1, p. 50–57, 1995.
- [9] MATLAB. *version 7.10.0 (R2010a)*. Natick, MassachusettsThe MathWorks Inc., 2010.

- [10] YE, W.; HEIDEMANN, J.; ESTRIN, D. An energy-efficient MAC protocol for wireless sensor networks. In: IEEE. *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings.* 2002. v. 3, p. 1567–1576.
- [11] IEEE Standard for Local and metropolitan area networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs). *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*, p. 1–314, Sept 2011.
- [12] HUANG, P.; XIAO, L.; SOLTANI, S.; MUTKA, M. W.; XI, N. The evolution of MAC protocols in wireless sensor networks: A survey. *Communications Surveys & Tutorials*, IEEE, v. 15, n. 1, p. 101–120, 2013.
- [13] 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, 2012.
- [14] RAJENDRAN, V.; OBRACZKA, K.; GARCIA-LUNA-ACEVES, J. J. Energy-efficient, collision-free medium access control for wireless sensor networks. *Wireless Networks*, Springer-Verlag New York, Inc., v. 12, n. 1, p. 63–78, 2006.
- [15] LEVIS, P.; MADDEN, S.; POLASTRE, J.; SZEWCZYK, R.; WHITEHOUSE, K.; WOO, A.; GAY, D.; HILL, J.; WELSH, M.; BREWER, E. et al. TinyOS: An operating system for sensor networks. In: *Ambient intelligence*. Springer, 2005. p. 115–148.
- [16] Módulos sem fio Rene. Acessado em: 05/01/2014. Disponível em: <www.xbow.com>.
- [17] HAN, K.; LUO, J.; LIU, Y.; VASILAKOS, A. V. Algorithm design for data communications in duty-cycled wireless sensor networks: A survey. *Communications Magazine, IEEE*, IEEE, v. 51, n. 7, 2013.
- [18] MORAES, R. M. de; SADJADPOUR, H. Wireless Network Protocols. In: *Jerry D. Gibson. (Org.). Mobile Communications Handbook*. 3th ed.. ed. CRC press, 2012. p. 603–614.
- [19] YE, W.; HEIDEMANN, J.; ESTRIN, D. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *Networking, IEEE/ACM Transactions on*, IEEE, v. 12, n. 3, p. 493–506, 2004.
- [20] HILL, J. L.; CULLER, D. E. Mica: A wireless platform for deeply embedded networks. *Micro*, IEEE, v. 22, n. 6, p. 12–24, 2002.
- [21] University of California, Berkeley. Mica2. Acessado em: 07/06/2014. Disponível em: <<http://www.eol.ucar.edu/isf/facilities/isa/internal/CrossBow/DataSheets/mica2.pdf>>.
- [22] DAM, T. V.; LANGENDOEN, K. An adaptive energy-efficient MAC protocol for wireless sensor networks. In: ACM. *Proceedings of the 1st international conference on Embedded networked sensor systems*. 2003. p. 171–180.
- [23] Módulos sem fio eyes. Acessado em: 02/01/2014. Disponível em: <<http://eyes.eu.org/sensnet.htm/>>.

- [24] EL-HOIYDI, A.; DECOTIGNIE, J.-D. WiseMAC: an ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks. In: IEEE. *Computers and Communications, 2004. Proceedings. ISCC 2004. Ninth International Symposium on*. 2004. v. 1, p. 244–251.
- [25] POLASTRE, J.; HILL, J.; CULLER, D. Versatile low power media access for wireless sensor networks. In: ACM. *Proceedings of the 2nd international conference on Embedded networked sensor systems*. 2004. p. 95–107.
- [26] BUETTNER, M.; YEE, G. V.; ANDERSON, E.; HAN, R. X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks. In: ACM. *Proceedings of the 4th international conference on Embedded networked sensor systems*. 2006. p. 307–320.
- [27] MOSS, D.; LEVIS, P. BoX-MACs: Exploiting physical and link layer boundaries in low-power networking. *Technical Report SING-08-00, Computer Systems Laboratory Stanford University*, 2008.
- [28] DUTTA, P.; DAWSON-HAGGERTY, S.; CHEN, Y.; LIANG, C.-J. M.; TERZIS, A. Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless. In: ACM. *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*. 2010. p. 1–14.
- [29] SHA, M.; HACKMANN, G.; LU, C. Energy-efficient low power listening for wireless sensor networks in noisy environments. In: ACM. *Proceedings of the 12th international conference on Information processing in sensor networks*. 2013. p. 277–288.
- [30] University of California, Berkeley. TinyOS CVS Repository at SourceForge. Acessado em: 20/09/2013. Disponível em: <<http://sf.net/projects/tinyos/>>.
- [31] Telos - Ultra Low Power IEEE 802.15.4 Compliant Wireless Sensor Module, Datasheet, MoteIV Corporation. Acessado em: 03/01/2014. Disponível em: <<http://www.memsic.com.php5-12.dfw1-1.websitetestlink.com/products/wireless-sensor-networks/wireless-modules.html>>.
- [32] LU, G.; KRISHNAMACHARI, B.; RAGHAVENDRA, C. S. An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks. In: IEEE. *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*. 2004. p. 224.
- [33] NOROUZI, A.; ZAIM, A. H. et al. An integrative comparison of energy efficient routing protocols in wireless sensor network. *Wireless Sensor Network*, Scientific Research Publishing, v. 4, n. 03, p. 65, 2012.
- [34] CHEN, Z.; KHOKHAR, A. Self organization and energy efficient TDMA MAC protocol by wake up for wireless sensor networks. In: IEEE. *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*. 2004. p. 335–341.
- [35] CUNHA, F. D.; CUNHA, I.; WONG, H. C.; LOUREIRO, A. A.; OLIVEIRA, L. B. ID-MAC: An identity-based MAC protocol for Wireless Sensor Networks. In: IEEE. *Computers and Communications (ISCC), 2013 IEEE Symposium on*. 2013. p. 975–981.

- [36] Ns é um simulador de eventos discretos destinado a pesquisas de redes cabeadas e sem fio. Acessado em: 23/07/2013. Disponível em: <http://nsgm.isi.edu/nsgm/index.php/User_Information>.
- [37] QIN, F.; MITCHELL, J. E. AS-MAC: Utilizing the adaptive spreading code length for wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, ACM, v. 10, n. 1, p. 1, 2013.
- [38] GUIMARAES, V. G.; BAUCHSPIESS, A.; MORAES, R. M. de. Dynamic Timed Energy Efficient and data collision free MAC protocol for wireless sensor networks. In: IEEE. *Communications (LATINCOM), 2013 IEEE Latin-America Conference on*. Santiago, Chile, 2013. p. 1-6.
- [39] Módulos sem fio MICA. Acessado em: 07/06/2014. Disponível em: <<http://pt.scribd.com/doc/53913252/MICA-Data-Sheet>>.
- [40] IRIS - A 2.4 GHz Mote module used for enabling low-power, wireless sensor networks. Acessado em: 10/01/2014. Disponível em: <<http://www.memsic.com.php5-12.dfw1-1.websitetestlink.com/products/wireless-sensor-networks/wireless-modules.html>>.
- [41] Módulos sem fio MICAZ. Acessado em: 15/09/2013. Disponível em: <http://www.memsic.com/userfiles/files/Datasheets/WSN/micaz_datasheet-t.pdf>.
- [42] Módulos sem fio Zigbit. Acessado em: 03/06/2013. Disponível em: <http://www.atmel.com/images/atmel-8226-zigbit-2.4ghz-wireless-modules-atzb-24-a2b0_datasheet.pdf>.
- [43] Manual da linguagem nesC. Acessado em: 17/09/2013. Disponível em: <<http://www.tinyos.net/dist-2.0.0/tinyos-2.0.0beta1/doc/nesc/ref.pdf>>.
- [44] Desenvolvedora do Bitcloud. Acessado em: 09/06/2014. Disponível em: <<http://www.atmel.com/default.aspx>>.
- [45] ATMEL. Sistema operacional Bitcloud. Acessado em: 07/06/2013. Disponível em: <<http://www.atmel.com/tools/bitcloud-zigbee-pro.aspx>>.
- [46] ATMEL. Pacote de comunicação sem fio aberto IEEE 802.15.4 MAC. Acessado em: 08/11/2013. Disponível em: <http://www.atmel.com/tools/ieee802_15_4mac.aspx>.
- [47] KERNIGHAN, B. W.; RITCHIE, D. M.; EJEKLINT, P. *The C programming language*. prentice-Hall Englewood Cliffs, 1988.
- [48] CUNHA, A.; SEVERINO, R.; PEREIRA, N.; KOUBÂA, A.; ALVES, M. Zigbee over tinyOS: implementation and experimental challenges. CISTER, 2008.
- [49] HAVERKORT, B. R. Performance of Computer Communication Systems: A Model-Based Approach. John Wiley & Sons, Inc., 1998.

- [50] Agilent Technologies. 34410A Multímetro digital de 6 1/2 dígitos e alta performance. Acessado em: 04/07/2014. Disponível em: <<http://www.home.agilent.com/pt/pd-692834-pn-34410A/digital-multimeter-6-digit-high-performance>>.
- [51] SENSIRION. (2011) Datasheet SHT7x (SHT71, SHT75) Humidity and Temperature Sensor IC, Version 5. Acessado em: 16/06/2014. Disponível em: <http://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/Humidity/Sensirion_Humidity_SHT7x_Datasheet_V5.pdf>.
- [52] The E.R.I.K.A. Enterprise realtime Operating System. Acessado em: 11/06/2014. Disponível em: <<http://www.evidence.eu.com/>>.
- [53] PAGANO, P.; CHITNIS, M.; ROMANO, A.; LIPARI, G.; SEVERINO, R.; ALVES, M.; SOUSA, P. G.; TOVAR, E. ERIKA and open-ZB: an implementation for real-time wireless networking. In: ACM. *Proceedings of the 2009 ACM symposium on Applied Computing*. 2009. p. 1687–1688.

ANEXOS

