



GUSTAVO FIGUEIREDO ARAÚJO

**PROTOCOLO MAC ANYCAST COM PREÂMBULOS
AVANÇADOS PARA REDES DE SENSORES SEM FIO**

LAVRAS – MG

2017

GUSTAVO FIGUEIREDO ARAÚJO

**PROTOCOLO MAC ANYCAST COM PREÂMBULOS AVANÇADOS PARA REDES
DE SENSORES SEM FIO**

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

Dr. Tales Heimfarth

Orientador

Dr. João Carlos Giacomini

Coorientador

LAVRAS – MG

2017

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

Araújo, Gustavo Figueiredo.

Protocolo MAC anycast com preâmbulos avançados para redes
de sensores sem fio / Gustavo Figueiredo Araújo. - 2017.

87 p. : il.

Orientador(a): Tales Heimfarth.

Coorientador(a): João Carlos Giacomin.

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

Bibliografia.

1. Redes de Sensores sem Fio. 2. Protocolo MAC. 3.
Comunicação Anycast. I. Heimfarth, Tales . II. Giacomin, João
Carlos. III. Título.

GUSTAVO FIGUEIREDO ARAÚJO

**PROTOCOLO MAC ANYCAST COM PREÂMBULOS AVANÇADOS PARA REDES
DE SENSORES SEM FIO**

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

APROVADA em 30 de maio de 2017.

Dr. Luiz Henrique Andrade Correia UFLA
Dr. Edison Pignaton de Freitas UFRGS

Dr. Tales Heimfarth
Orientador

Dr. João Carlos Giacomini
Co-Orientador

**LAVRAS – MG
2017**

Dedico esse trabalho a minha mãe Sandra, ao meu pai Luiz Carlos, as minhas irmãs Gabriela e Graciele, ao meu sobrinho Lucas e a minha noiva Isabel.

RESUMO

As redes de sensores sem fio empregam ciclos de trabalho para economizar energia, com o custo do aumento da latência no encaminhamento de mensagens. Protocolos *cross-layer* que utilizam comunicação *anycast*, conseguem uma redução de latência em RSSF assíncronas que operam em ciclos de trabalho. Uma série de preâmbulos é enviada de modo a encontrar o nó do próximo salto, que é selecionado a partir de um grupo de nós candidatos ao encaminhamento (FCS - *forwarding candidate set*). Este trabalho propõe um novo protocolo *cross-layer anycast* nominado APA-MAC (Advanced Preamble Anycast for MAC layer), para o encaminhamento de mensagens com baixa latência em redes de sensores sem fio que utilizam ciclos de trabalho. No protocolo APA-MAC, os preâmbulos se propagam à frente, criando a rota até o destino, enquanto que a mensagem de dados segue alguns saltos atrás. A transmissão de dados para o próximo salto ocorre mediante uma sinalização do receptor indicando que o canal está livre. Dessa forma, a transmissão dos preâmbulos não precisa ser interrompida se a mensagem de dados se aproximar da área de transmissão dos preâmbulos. Portanto, a propagação simultânea de preâmbulos e mensagem de dados, com a transmissão ininterruptível de preâmbulos, promovem ao APA-MAC um ganho na redução da latência. Além disso, baseado no tamanho da mensagem de dados, a cardinalidade do FCS é definida de modo que acelere a propagação dos preâmbulos, evitando que no momento da recepção dos dados, o nó encontre o canal ocupado. Os resultados da simulação mostraram que o protocolo proposto mantém baixo consumo energético, e alcança no mínimo 10% menor latência em relação aos protocolos MAC assíncronos avaliados, além de utilizar mecanismos que proporcionam uma taxa de mensagens entregues com sucesso de 98% em cenários propícios à colisões.

Palavras-chave: *cross-layer*, protocolo MAC assíncrono, *anycast*, redes de sensores sem fio

ABSTRACT

Wireless sensor networks use duty-cycling to save energy, with the cost of increasing latency in message routing due to additional hops. Cross-layer protocols that use anycast communication achieve a latency reduction in asynchronous WSNs that operate in duty-cycles. A series of preambles is sent in order to find the next hop, which is selected from a forwarding candidate set (FCS). This work proposes a new cross-layer protocol named APA-MAC (Advanced Preamble Anycast for MAC layer), for the delivery of messages with low latency in wireless sensors networks that employ duty-cycling. In the APA-MAC protocol, the preambles propagate ahead, creating a propagation path to the destination, and the data message follows this path few hops behind. The data transmission to the next hop is triggered by a signal indicating the vacancy of the channel. Due to this mechanism, transmissions of preambles do not have to be interrupted if the data message approaches the area. Therefore, the simultaneous propagation of preambles and data message, with the uninterrupted transmission of preambles, is responsible for latency reduction in the APA-MAC. In addition, based on the size of the data message, the cardinality of the FCS is defined in order to accelerate the propagation of the preambles, avoiding that at the time of receiving the data, the node finds the channel occupied. The results of the simulation showed that the proposed protocol maintains a low power consumption, achieving at least 10% less latency in relation to the other evaluated asynchronous MAC protocols. Moreover, the rate of successfully transmitted messages is about 98% in collisions prone scenarios.

Keywords: cross-layer, asynchronous MAC protocol, anycast, wireless sensor networks

LISTA DE SIGLAS

ACK	Acknowledgement
APA-MAC	Advanced Preamble Anycast for MAC layer
CTS	Clear-to-Send
DCF	Distributed Coordination Function
eACK	Early Acknowledgement
FCS	Forwarding Candidate Set
IEEE	Institute of Electrical and Electronics Engineers
ISO	International organization of Standardization
LLC	Logical Link Control
MAC	Media Access Control
MANET	Mobile Ad hoc Network
MEMS	Micro Electro Mechanical Systems
OSI	Open System Interconnection Reference Model
PAX-MAC	Preamble Ahead Crosslayer for MAC layer
QoS	Quality of Service
RSSF	Redes de Sensores sem Fio
RTS	Request-to-Send
SSCS	Service Specific Convergence Sublayer
TDMA	Time Division Multiple Access
UFLA	Universidade Federal de Lavras
WLAN	Wireless Local Area Network

LISTA DE FIGURAS

Figura 2.1 – Componentes de um nó sensor	18
Figura 2.2 – Pilha de protocolos do padrão IEEE 802.15.4	21
Figura 2.3 – Protocolos baseados em TDMA	24
Figura 2.4 – Protocolos com período ativo comum	25
Figura 2.5 – Protocolo assíncrono com amostragem de preâmbulo	26
Figura 2.6 – Representação esquemática de tipos de amostragem de preâmbulos	28
Figura 2.7 – Tipos de Transmissão de Dados	34
Figura 2.8 – Princípio dos protocolos <i>anycast</i>	37
Figura 3.1 – Visão geral do protocolo PAX-MAC	42
Figura 3.2 – Máquina de estados do protocolo PAX-MAC	45
Figura 4.1 – Visão geral do protocolo APA-MAC	52
Figura 4.2 – FCS Fixo	55
Figura 4.3 – FCS Fixo	55
Figura 4.4 – Otimização de saltos	56
Figura 4.5 – Mecanismo de Proteção do Canal	58
Figura 4.6 – Máquina de estados do protocolo APA-MAC	59
Figura 5.1 – Latência para diferentes configurações do FCS. Mensagens com tempo de transmissão igual a 15% do ciclo total	66
Figura 5.2 – Latência para diferentes configurações do FCS. Mensagens com tempo de transmissão igual a 50% do ciclo total	66
Figura 5.3 – Avaliação da latência quando origem e destino estão a 650 metros	69
Figura 5.4 – Avaliação da latência quando origem e destino estão a 1300 metros	70
Figura 5.5 – Avaliação do consumo energético quando origem e destino estão a 650 metros	72
Figura 5.6 – Avaliação do consumo energético quando origem e destino estão a 1300 metros	73
Figura 5.7 – Esquema do percurso dos fluxos concorrentes	74
Figura 5.8 – Avaliação da latência com fluxos concorrentes	75
Figura 5.9 – Avaliação da taxa de entrega com fluxos concorrentes	76
Figura 5.10 – Terminal Oculto	77

LISTA DE QUADROS

Quadro 2.1 – Consumo energético do transceptor	18
Quadro 2.2 – Consumo energético do microcontrolador	19
Quadro 3.1 – Lista dos estados da máquina de estados	45
Quadro 3.2 – Lista das transições dos estados da máquina de estados	46
Quadro 4.1 – Lista dos estados da máquina de estados	59
Quadro 4.2 – Lista das transições dos estados da máquina de estados	60
Quadro 5.1 – Parâmetros usados na simulação	65

SUMÁRIO

1	Introdução	10
1.1	Definição do Problema	11
1.2	Motivação	12
1.3	Objetivo Geral	13
1.4	Objetivos Específicos	14
1.5	Contribuições do Trabalho	14
1.6	Estrutura do Trabalho	14
2	REFERENCIAL TEÓRICO	15
2.1	Redes de Sensores e Atuadores	15
2.2	Fatores que Influenciam no Projeto de uma Rede de Sensores sem Fio	17
2.3	Arquitetura de Comunicação	20
2.4	Camada MAC	22
2.5	Camada de Rede	30
2.5.1	Tipos de Protocolos da Camada de Rede para RSSF	32
2.5.2	Tipos de Transmissão de Dados	34
2.6	Abordagem <i>Cross-Layer</i>	34
2.7	Comunicação <i>Anycast</i>	36
3	Trabalhos Relacionados	38
3.1	PAX-MAC	40
3.1.1	Seleção do FCS	42
3.1.2	Colisão Iminente	43
3.1.3	Máquina de estados	44
4	APA-MAC	48
4.1	Visão Geral	48
4.2	Critério de Seleção do FCS	53
4.3	Configuração da Cardinalidade do FCS	54
4.4	Otimização de saltos	56
4.4.1	Proteção do canal	57
4.4.2	Máquina de estados	58
5	Resultados e Discussões	63
5.1	Configurações da Simulação	63

5.2	Avaliação da Latência	65
5.3	Avaliação do Consumo Energético	71
5.4	Avaliação da Latência com Fluxos Concorrentes	72
5.5	Avaliação da Taxa de Entrega com Fluxos Concorrentes	75
6	Conclusões e Trabalhos Futuros	78
	REFERÊNCIAS	80

1 Introdução

Os avanços tecnológicos dos sistemas microeletromecânicos (MEMS - *Micro Electro Mechanical Systems*), da comunicação sem fio e da eletrônica digital, tornou técnica e economicamente viável, a fabricação de dispositivos sensores multifuncionais de pequenas dimensões, com baixo custo e baixo consumo energético (HUANG et al., 2013; YANG, 2013). Estes dispositivos sensores, conhecidos como nós sensores, são capazes de sensorear, medir e coletar informações do ambiente no qual estão inseridos. Os nós sensores trabalham cooperativamente, encaminhando estas informações a outros nós sensores, através de enlaces sem fio, até que a mensagem chegue ao destino, formando assim, uma rede de sensores sem fio (RSSF).

A tecnologia da rede de sensor sem fio oferece inúmeras vantagens em relação às soluções de redes convencionais, tais como, custos mais baixos, escalabilidade, flexibilidade e facilidade de implantação, permitindo seu uso em uma ampla gama de aplicações em áreas como industriais, militares, ambientais, de saúde, automação residencial, entre muitas outras. Segundo Rawat et al. (2014) em 2014 o mercado das RSSF movimentou cerca de US\$1 bilhão e em 2022, a previsão é que o mercado movimente US\$2 bilhões.

De acordo com Cano et al. (2011) e Kredo e Mohapatra (2007), uma RSSF pode ser formada por centenas ou milhares de nós sensores. Portanto, o custo de um único nó sensor é muito importante para justificar o custo total da rede. Além disso, os nós sensores podem ser implantados em ambientes ou sob circunstâncias que inviabilizam o acesso aos nós após a implantação, os tornando descartáveis. Entretanto, manter um baixo custo coloca limites sobre as tecnologias que cada nó sensor pode utilizar.

Os nós sensores são alimentado por baterias, uma vez que uma fonte de energia tradicional, como uma rede elétrica, pode não estar disponível, ou até mesmo ser inviável. Portanto, além dos componentes do *hardware*, a fonte de energia dos nós sensores também é limitada.

Espera-se que os nós sensores funcionem por um longo tempo sem intervenção humana, portanto todas as ações devem ser tomadas da forma mais eficiente em termos de energia. Esta é a primeira preocupação ao desenvolver *softwares* para RSSF, incluindo os protocolos da rede.

Em relação aos protocolos de roteamento, uma abordagem interessante é o uso de informações geográficas considerando a estimativa da posição dos nós. Esta abordagem reduz o esforço de coordenação para manter o conhecimento atualizado da topologia da rede (CASARI et al., 2006).

Considerando os protocolos da camada MAC, a principal estratégia para economizar energia é o uso de ciclos de trabalho (*duty-cycling*) (BUETTNER et al., 2006). Nesta estratégia, os nós são mantidos em um modo de baixo consumo de energia na maior parte do tempo (estado inativo). Periodicamente, os nós são reativados (estado ativo) para sensoriar variáveis do ambiente e transmitir os dados adquiridos, retornando ao estado inativo após um determinado intervalo de tempo.

Segundo (TAN; CHAN, 2010), estudos de desempenho mostram que, os ciclos de trabalho são eficazes na redução do consumo de energia nas RSSF, devido às características de tráfego esporádico do nó sensor. Por exemplo, um ciclo de trabalho de 1% pode potencialmente reduzir o consumo de energia de uma rede em 99% quando nenhum tráfego está sendo gerado. Entretanto, este método induz uma latência devido aos períodos inativos dos nós sensores. Desta forma, um grande esforço de pesquisa tem sido realizado tentando reduzir a latência introduzida pelos ciclos de trabalho (CARRANO et al., 2014; HUANG et al., 2013).

1.1 Definição do Problema

Apesar da técnica de ciclos de trabalho ser uma excelente estratégia para estender a vida da RSSF, este método induz uma latência devido aos períodos inativos dos nós sensores. Antes de enviar uma mensagem, o nó tem que esperar até que o receptor (o nó do próximo salto) esteja ativo, acumulando uma latência a cada salto realizado até o destino.

Segundo Carrano et al. (2014) e Huang et al. (2013), várias pesquisas têm sido realizadas a fim de reduzir a latência causada pelos ciclos de trabalho. Entre as estratégias para lidar com este problema, a mais eficaz é cuidadosamente escalonar os tempos de atividade de todos os nós ao longo do caminho até o destino da mensagem. Este método é explorado em RMAC (DU; SAHA; JOHNSON, 2007), DMAC (LU; KRISHNAMACHARI; RAGHAVENDRA, 2004) e EX-MAC (HONG; KIM, 2009). No entanto, todos eles enfrentam a dificuldade de manter o sincronismo entre os nós sensores, o que é um problema comum em RSSF.

Além das abordagens tradicionais, abordagens *cross-layer* têm sido propostas a fim de tornar as RSSF mais eficientes no roteamento de mensagens, reduzindo assim o consumo de energia (RAULT; BOUABDALLAH; CHALLAL, 2014; TOKLU; ERDEM, 2014; VURAN; AKYILDIZ, 2010). A arquitetura *cross-layer* explora a possibilidade de obter e/ou alterar parâmetros de duas ou mais camadas, a fim de atingir um objetivo de otimização (MENDES; RODRIGUES, 2011; ZUO et al., 2014).

A integração de funcionalidades da camada MAC e da camada de rede (roteamento) foram exploradas por vários projetos na última década conforme em Mendes e Rodrigues (2011). Entre esses projetos, uma abordagem *cross-layer* eficiente é a comunicação *anycast* (ASHRAF; VAIDYA; KRAVETS, 2011; EL-HOYDI; DECOTIGNIE, 2004; KIM et al., 2010; TAN; CHAN, 2010; ZENG et al., 2007), uma classe de protocolos que tira proveito da característica de redundância inerente das RSSF densas, para reduzir o atraso causado pelos ciclos de trabalho.

Diferentemente dos protocolos de comunicação *unicast* tradicionais, protocolos *anycast* não escolhem um único nó para a retransmissão, mas sim, um subconjunto de nós vizinhos com condições apropriadas para encaminhar a mensagem. Os protocolos *anycast* existentes normalmente unificam a funcionalidade de protocolos MAC assíncronos, como X-MAC (BUETTNER et al., 2006) e WiseMAC (EL-HOYDI; DECOTIGNIE, 2004) com protocolos de roteamento oportunistas (KIM et al., 2010; ZENG et al., 2007). Segundo Ashraf, Vaidya e Kravets (2011), adicionar um modo de comunicação *anycast* a um protocolo MAC assíncrono *unicast* já existente pode reduzir a latência em até 30% quando há dois nós candidatos a realizar o encaminhamento da mensagem

1.2 Motivação

Observando outras possibilidades de aumentar as vantagens obtidas com os protocolos *anycast*, em Heimfarth et al. (2016) foi proposto o protocolo PAX-MAC (*Preamble Ahead Crosslayer MAC*), sendo este, um protocolo *cross-layer* assíncrono que realiza o avanço de preâmbulos ao mesmo tempo em que ocorre a transmissão da mensagem de dados. Melhorias na redução da latência são providas explorando os conceitos de roteamento oportunista e a comunicação *anycast*. O protocolo introduz um esquema de reserva de canal para a transmissão dos dados em uma RSSF assíncrona, escalonando os nós sensores de uma maneira similar à empregada em Du, Saha e Johnson (2007), Hong e Kim (2009), Kechar et al. (2008) e Sun et al. (2008). No entanto, diferentemente das abordagens do estado da arte, o PAX-MAC não necessita de energia para manter a sincronização como Du, Saha e Johnson (2007), Kechar et al. (2008) e Sun et al. (2008), e, ao mesmo tempo, não se restringe ao tráfego em rajadas como Hong e Kim (2009). PAX-MAC consegue resultados próximos de uma rede onde os nós estão sempre ativos, mantendo o perfil de energia de protocolos que usam ciclos de trabalho. Isso é feito sem qualquer sincronização dispendiosa entre os nós. Segundo Heimfarth et al. (2016), o

protocolo obteve no mínimo 20% de redução na latência, considerando diferentes tamanhos de mensagens, quando comparado com protocolos *anycast* da literatura.

A abordagem empregada em Heimfarth et al. (2016) para protocolos MAC assíncronos, é baseada na união de duas estratégias que se mostram eficientes na redução da latência causada pelos ciclos de trabalho. A primeira é escalonar os tempos de atividade de todos os nós ao longo do caminho até o destino da mensagem. A segunda é aproveitar a redundância de caminhos, isto é, vários nós candidatos disponíveis para realizar a retransmissão da mensagem. A união destas estratégias se mostrou eficiente na redução da latência.

No protocolo PAX-MAC, o preâmbulo propaga avançando pelos nós transmissores de modo a criar uma reserva de tempo para a recepção da mensagem de dados que virá, realizando o avanço de preâmbulos ao mesmo tempo em que ocorre a transmissão de dados. Esta abordagem promove redução na latência, mas o protocolo possui algumas características que limitam seu desempenho. Nesta abordagem, os preâmbulos seguem à frente da mensagem de dados, mas caso ocorra a aproximação da mensagem de dados ao preâmbulo, para evitar uma colisão, PAX-MAC suspende o envio de preâmbulos até a chegada da mensagem de dados. Entretanto, esta estratégia prejudica o ganho na latência, uma vez que avanço dos preâmbulos é temporariamente interrompido até que o nó que estava enviando os preâmbulo receba a mensagem de dados. Para minimizar este problema, o nó de origem mantém um atraso no envio da mensagem de dados, reduzindo as possibilidades que a mensagem aproxime-se do preâmbulos. Entretanto, se não houver uma redução na velocidade média no avanço dos preâmbulos, que justifique o tempo definido para esse atraso, ocorrerá o aumento na latência total para a entrega de dados.

Embora as estratégias adotadas em Heimfarth et al. (2016) sejam interessantes, ainda possuem deficiências que podem ser exploradas a fim de obter melhores resultados. Portanto, ainda cabe novos estudos que possam elaborar estratégias ainda mais eficientes na redução da latência.

1.3 Objetivo Geral

O objetivo geral deste trabalho é desenvolver um protocolo MAC assíncrono para redes de sensores sem fio que proporcione melhorias na redução de latência, mantendo baixo consumo energético.

1.4 Objetivos Específicos

Para atingir o objetivo geral, foram determinados os seguintes objetivos específicos:

- Projetar o protocolo baseado nas estratégias estudadas
- Desenvolver o protocolo no simulador *GrubiX Wireless Network Simulator*;
- Programar alguns dos principais protocolos MAC assíncronos que utilizam abordagem *anycast* no simulador;
- Realizar a avaliação de desempenho, comparando os resultados com os obtidos dos protocolos implementados.

1.5 Contribuições do Trabalho

A contribuição deste trabalho é a elaboração de técnicas eficazes para reduzir a latência causada pelo ciclo de trabalho em RSSF assíncronas. Assim como, o desenvolvimento e avaliação de um protocolo MAC assíncrono que adota estas técnicas.

1.6 Estrutura do Trabalho

Este trabalho está organizado da seguinte forma: os conceitos sobre redes de sensores, arquiteturas, classes de protocolos, abordagens *cross-layer*, comunicação *anycast* e trabalhos relacionados são apresentados no Capítulo 2. O protocolo proposto, é apresentado no capítulo 3. No capítulo 4 são apresentadas as configurações das simulações, assim como os resultados e discussões. Por fim, o Capítulo 5 apresenta as considerações finais deste trabalho.

2 REFERENCIAL TEÓRICO

Neste capítulo são apresentados conceitos de redes sensores sem fio, assim como a organização das camadas do padrão IEEE 802.15.4, e as diferentes abordagens em relação às camadas MAC e de rede. Além disso, são apresentadas duas abordagens que possuem o objetivo de maximizar o desempenho global da rede e minimizar o gasto de energia. A primeira abordagem, denominada *cross-layer*, quebra algumas regras de comunicação da arquitetura, permitindo a comunicação de camadas não adjacentes ou o compartilhamento de variáveis entre as camadas. A segunda abordagem, denominada *anycast*, aproveita a redundância do caminho, isto é, vários nós candidatos disponíveis para realizar a retransmissão da mensagem.

2.1 Redes de Sensores e Atuadores

As redes de sensores sem fio (RSSF) podem ser vistas como um tipo especial de rede móvel ad hoc (MANET - *Mobile Ad hoc Network*). Uma rede ad hoc é uma rede local sem fios (WLAN - *Wireless Local Area Network*) onde os dispositivos móveis são parte da rede somente quando eles estão suficientemente próximos, de forma que seja possível realizar transmissões. Não há infra-estrutura fixa e a informação é passada ponto a ponto através de múltiplos saltos (*multihop*).

As RSSF são focadas na conectividade sem a necessidade, geralmente, de cabeamento para conectar sensores e atuadores. O funcionamento dessa rede destina-se a coletar informações dos sensores e enviar comandos de controle para os atuadores através da rede sem fio.

Segundo Yang (2013), redes de sensores e atuadores já existem há décadas. Sistemas de controle baseados em computadores são tipicamente redes de sensores e atuadores cabeados. Os sensores e atuadores são conectados a um computador central ou um terminal de controle que implementa funções de monitoramento e controle. Este tipo de rede de sensores com fio é muitas vezes vista no controle industrial, tais como controle de processo e controle de produção. Por causa do envolvimento de grande quantidade de cabeamento na instalação, o custo da instalação de redes de sensores cabeada é elevado, seja por custos de cabos, material, trabalho, teste ou verificação. Além disso, cabos requerem o uso de conectores que podem se soltar, estar mal conectados ou até mesmo quebrar.

O uso de um grande número de sensores cabeados em rede traz uma considerável complexidade ao sistema, incluindo a instalação de cabos, fonte de energia e configuração, sendo

em muitos casos, impossível sua implantação, principalmente em áreas remotas, inseguras, de difícil acesso ou de grande extensão.

Os circuitos integrados e sistemas micro-eleto-mecânicos evoluíram ao ponto de permitir a integração de comunicações sem fio, sensores e processamento de sinais juntos em um único dispositivo de baixo custo, chamado de nó sensor (YANG, 2013; HUANG et al., 2013). A interligação destes nós sensores forma uma rede de sensores sem fio.

Os circuitos de sensoriamento medem as condições ambientais ao redor do sensor, e os transforma em sinais elétricos que podem ser processados. Os nós sensores são espacialmente distribuídos, e trabalham cooperativamente, enviando as informações processadas para outros nós utilizando sinais de rádio. Estas informações se propagam através dos nós sensores até alcançarem o destino pré-definido. Para facilitar a implantação da rede de nós sensores, cada nó é alimentado por baterias, uma vez que uma fonte de energia tradicional, como uma rede elétrica, pode não estar disponível.

A tecnologia da rede de sensor sem fio oferece inúmeras vantagens em relação às soluções de rede convencionais, tais como, custos mais baixos, escalabilidade, flexibilidade e facilidade de implantação, permitindo seu uso em uma ampla gama de aplicações. Com os avanços da tecnologia e os nós sensores ficando cada vez mais inteligentes, menores e mais baratos, bilhões de sensores sem fio estão sendo implantados em inúmeras aplicações. Com várias aplicações e oportunidades de negócios que surgem todos os dias, o mercado de RSSF está previsto para aumentar de US\$1 bilhão em 2014 para US\$2 bilhões em 2022 (RAWAT et al., 2014).

Segundo Rault, Bouabdallah e Challal (2014), Rawat et al. (2014), alguns potenciais domínios de aplicação das redes sensores sem fio são: na área militar, ambiental, na assistência médica e na segurança. Para uso militar, os nós sensores podem ser usados para detectar, localizar ou rastrear movimentos de inimigos. Em caso de desastres naturais, os nós sensores podem sensoriar o ambiente para anteciparem as catástrofes com antecedência. Na medicina, os nós sensores podem ajudar na monitoração da saúde do paciente. Na área de segurança, os sensores podem oferecer vigilância e aumentar o estado de alerta para potenciais ataques terroristas. Dependendo da aplicação e do tipo de sensoriamento a ser realizado, é possível encontrar diversos tipos de nós sensores sem fios.

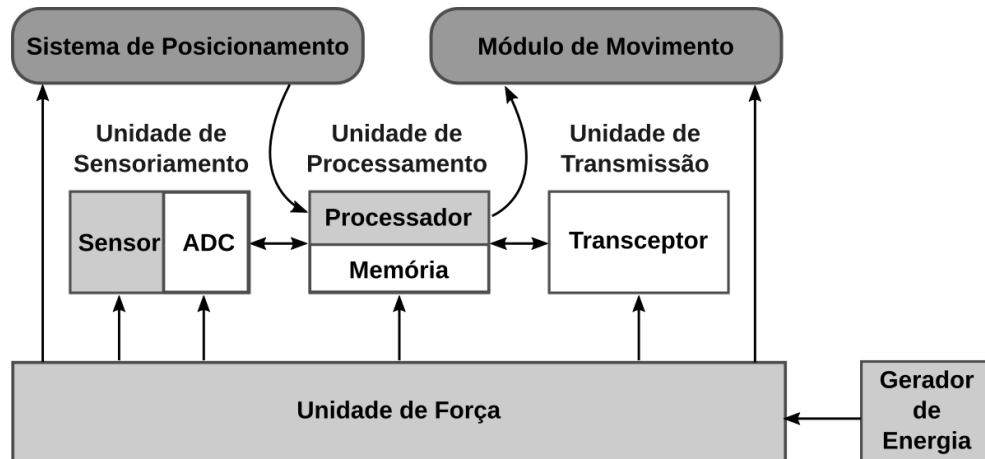
2.2 Fatores que Influenciam no Projeto de uma Rede de Sensores sem Fio

Um projeto de rede de sensores é influenciado por muitos fatores, o que inclui tolerância a falhas, escalabilidade, custos de produção, ambiente de operação, topologia da rede, restrições de hardware, meios de transmissão e consumo energético (YANG, 2013).

Restrições de hardware: Como mostrado na Figura 2.1, um nó sensor é constituído pelos seguintes componentes básicos: uma unidade de sensoriamento, uma unidade de processamento, uma unidade de transmissão e uma unidade de energia (YANG, 2013). Dependendo da aplicação, os nós sensores também podem ter componentes adicionais, tais como sistemas de localização, gerador de energia e mobilizadores. Unidades de sensoriamento são geralmente compostas por duas subunidades: sensores e conversores de sinais analógicos para digital. Os sinais analógicos, produzidos pelos sensores baseados no fenômeno observado, são convertidos em sinais digitais, e enviados para a unidade de processamento. A unidade de processamento, que é geralmente associada a uma unidade de armazenamento, executa os procedimentos para realizar as tarefas de sensoriamento atribuídas. A unidade de transmissão conecta o nó à rede. A unidade de alimentação é um dos componentes mais importantes de um nó sensor e pode suportar unidades de energia como células solares. Existem também outras subunidades, que são dependentes da aplicação. Em uma rede de sensores, a maioria das técnicas de roteamento e tarefas de sensoriamento requer o conhecimento da localização com uma alta precisão. Portanto, é comum que nós sensores tenham um sistema de posicionamento. Um módulo de movimento pode algumas vezes ser necessário para mover o nó sensor quando é requerido pelas tarefas atribuídas. Muitas vezes, estas subunidades precisam ser pequenas o suficiente para caber em um módulo do tamanho de uma caixa de fósforo.

Consumo energético: O nó sensor sem fio, sendo um dispositivo microeletrônico, só pode ser equipado com uma fonte de energia limitada. Em alguns cenários de aplicações, a reposição da fonte de energia pode ser impossível. O tempo de vida do nó sensor, mostra uma forte dependência com o tempo de vida da bateria. Em uma rede de sensores *ad hoc* de múltiplos saltos, cada nó desempenha o duplo papel de gerar e rotear dados. O não funcionamento de alguns nós pode causar significantes alterações topológicas, tornando-se necessário o reencaminhamento de pacotes e reorganização da rede. Assim, a conservação e o gerenciamento de energia assumem uma maior importância. É por estas razões

Figura 2.1 – Componentes de um nó sensor



que as pesquisas são direcionadas para projetos de protocolos com consciência energética para redes de sensores sem fio.

Como referência para o consumo energético dos nó sensores, os Quadros 2.1 e 2.2, segundo Carrano et al. (2014), representam respectivamente os dados do consumo energético do transceptor e do microcontrolador de seis tipos de nós sensores.

Quadro 2.1 – Consumo energético do transceptor

Nó Sensor	Transceptor			
	Modelo	Consumo Energético		
		Inativo ¹	Ocioso ²	Ativo (TX/RX) ³
Mica 2	TI CC1000	0,2 a 1 μ A	0,030 a 0,105mA	10 a 27mA
Mica2dot	TI CC1000	0,2 a 1 μ A	0,030 a 0,105mA	10 a 27mA
MicaZ	TI CC2420	20 μ A	0,426mA	11 a 20mA
TelosB	TI CC2420	20 μ A	0,426mA	11 a 20mA
Imote2	TI CC2420	20 μ A	0,426mA	11 a 20mA
Iris	Atmel AT86RF230	0,02 μ A	1,5mA	10 a 17mA

¹ No estado inativo (*sleep state*), o regulador de tensão é ligado, mas o cristal oscilador não está ativado.

² Uma transição do estado inativo para ocioso é concluída quando o cristal oscilador é ligado. No estado ocioso, o rádio está pronto para assumir o estado de transmissão ou recepção.

³ O consumo energético durante a transmissão depende da potência de transmissão e, para o modelo CC1000, também da frequência utilizada (433 ou 868 MHz). O consumo energético durante os estados de recepção, é tipicamente da mesma ordem de grandeza dos estados de transmissão.

Dentre os componentes de hardware que são essenciais para a manutenção da conectividade e formação das RSSF, o transceptor é o componente com maior consumo energético. Ele consome energia mesmo quando está ligado em modo de recepção sem que haja comunicação.

Tolerância a falhas: Alguns nós sensores podem falhar ou ser bloqueados devido à falta de energia, ter danos físicos ou interferências do ambiente. A falha dos nós sensores não

Quadro 2.2 – Consumo energético do microcontrolador

Nó Sensor	Microcontrolador			Fonte de energia
	Modelo	Consumo Energético		
		Inativo	Ativo	
Mica 2	ATmega128L	15 μ A	8mA	2xAA pilhas
Mica2dot	ATmega128L	15 μ A	8mA	3v bateria
MicaZ	ATmega128L	15 μ A	8mA	2xAA pilhas
TelosB	TI MSP430	5,1 μ A	1,8mA	2xAA pilhas
Imote2	Marvell PXA271	390 μ A	66mA	3xAA pilhas
Iris	ATmega1281	8 μ A	8mA	2xAA pilhas

deve afetar a tarefa global da rede de sensores. A tolerância a falhas é a capacidade de manter as funcionalidades da rede sem interrupção, mesmo quando ocorrem falhas em alguns nós sensores.

Escalabilidade: Dependendo da aplicação, o número de nós sensores em uma RSSF pode ser da ordem de centenas ou milhares. Os novos projetos devem ser capazes de trabalhar com estes números de nós. A densidade pode variar de alguns nós sensores para algumas centenas de nós sensores em uma região, que pode ser inferior a 10 metros de diâmetro.

Topologia da rede: O simples fato que os nós sensores estão frequentemente propensos a falhar, faz com que a manutenção da topologia seja uma tarefa desafiadora. Centenas de milhares de nós são implantados em todo o campo de sensoriamento. A implantação de um elevado número de nós requer um tratamento cuidadoso na manutenção de topologia.

Ambiente: Nós sensores são densamente implantados perto ou diretamente em contato com os fenômenos a serem observados. Portanto, devem ser projetados para trabalhar sem a intervenção humana e em áreas geográficas remotas. Os nós sensores podem, por exemplo, ser usados para trabalhar sob alta pressão no fundo do oceano, sob baixas e altas temperaturas, como no motor de um avião ou em áreas árticas, e em ambientes que causam interferência na comunicação.

Meios de transmissão: Em uma rede de sensores de múltiplos saltos, a comunicação entre os nós é feita por meio sem fio. Estas ligações podem ser formadas por rádio, infravermelho ou mídia óptica. Para permitir o funcionamento global destas redes, o meio de transmissão escolhido também deve estar disponível em outros países.

Custos de produção: Uma vez que as redes de sensores podem ter um grande número de nós sensores, o custo de um único nó é muito importante para justificar o custo total da rede. Se o custo da rede é maior do que a implantação de sensores tradicionais, a rede de sensores sem fio não é economicamente justificável. Portanto, o custo de cada nó sensor deve-se manter o menor possível.

2.3 Arquitetura de Comunicação

O objetivo dos engenheiros de RSSF, é desenvolver soluções de rede sem fio economicamente viáveis, baseadas em padrões, que promovam baixo consumo de energia, e garantam segurança e confiabilidade para redes com baixa taxa de transmissão.

O projeto do *software* embarcado das redes de sensores sem fio, precisa contar com alguns padrões para garantir que o sistema de rede seja funcional em diferentes plataformas de hardware. Os padrões podem ser simplesmente divididos em duas categorias, públicos ou privados, de acordo com o propósito do projeto (YANG, 2013).

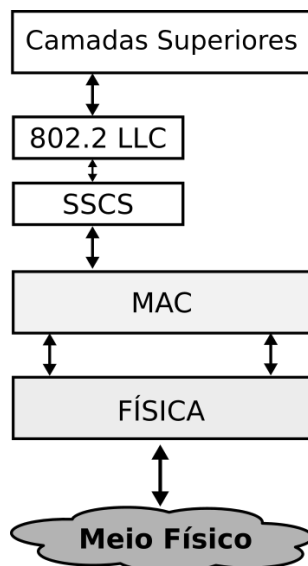
Os padrões privados geralmente são empregados em aplicações específicas, o que pode reduzir o suporte em outros requisitos. Por exemplo, se um padrão permite prolongar a vida útil da rede, o suporte para vazão de dados pode ser ineficiente.

Os padrões públicos têm um desempenho mais equilibrado que os padrões privados, uma vez que tendem a adotar o maior número possível de atributos dos fabricantes. Qualquer desenvolvimento de um padrão público, levará em consideração muitos aspectos possíveis para garantir a máxima compatibilidade. Padrões privados possuem um progresso de desenvolvimento mais rápido do que os padrões públicos, uma vez que só precisam aperfeiçoar o padrão para seu próprio propósito. No entanto, como indicado pelo seu nome, os padrões privados podem não estar disponíveis para acesso público.

Em 2003, foi formalmente ratificado pelo IEEE (*Institute of Electrical and Electronics Engineers*), a primeira versão do padrão público IEEE 802.15.4 (IEEE STANDARDS ASSOCIATION, 2015), que especifica a camada física e o controle de acesso ao meio para redes sem fio, com baixa taxa de transmissão, baixo consumo energético e de baixo alcance. É a base para as especificações ZigBee (FARAHANI, 2011), ISA100.11a (INTERNATIONAL SOCIETY OF AUTOMATION, 2017), 6LowPAN (SHELBY; BORMANN, 2011), WirelessHART (FIELD-COMM GROUP, 2017), e MiWi (MICROCHIP, 2017), cada uma das quais estende ainda mais o padrão através do desenvolvimento de camadas superiores que não são definidas pelo padrão

IEEE 802.15.4. Segundo Yang (2013), a Figura 2.2 ilustra a pilha de protocolos do padrão IEEE 802.15.4.

Figura 2.2 – Pilha de protocolos do padrão IEEE 802.15.4



A arquitetura consiste na definição de duas camadas, física e MAC. A camada física inclui principalmente o transceptor de rádio e seu mecanismo de controle de baixo nível. A camada MAC fornece as definições para a transferência de dados, acessando a camada física. A subcamada de convergência de serviços específicos (*SSCS - Service Specific Convergence Sublayer*) e o controle do link lógico (*LLC - Logical Link Control*) define um padrão para a comunicação com as camadas superiores, que podem ser definidas por outros padrões. Devido à característica de recursos limitados, as aplicações de redes de sensores sem fio normalmente requerem que o protocolo utilizado seja o mais simples possível, o que pode reduzir a sobrecarga do sistema. A arquitetura IEEE 802.15.4 é simples e permite que os desenvolvedores criem o software para a aplicação em um nível mais baixo, que possa interagir diretamente com a transferência de dados.

Outros padrões tradicionais que estejam em conformidade com modelo de referência ISO/OSI (*International organization of Standardization/Open System Interconnection Reference Model*), podem ser capazes de fornecer um serviço confiável, mas o modelo de sete camadas faz com que esse tipo de arquitetura seja complicado para ser aplicado nas RSSFs.

2.4 Camada MAC

A camada MAC é responsável por coordenar transmissões para um canal compartilhado, definindo como e quando um nó tentará realizar a transmissão. O procedimento de acesso ao meio pode basear-se numa atribuição de escalonamento (que pode ser fixo durante toda a vida útil da rede ou durante um determinado período de tempo) ou, pode ser baseada em um procedimento aleatório, em que o momento da tentativa de transmissão é decidido de forma independente em cada nó. Além disso, é possível definir abordagens híbridas que combinam ambas as técnicas.

Apesar do constante aumento das capacidades de processamento, memória e sincronização dos nós sensores, o projeto do protocolo MAC enfrenta a limitação energética, que se agrava, principalmente, quando o nó possui componentes com maior capacidade ou maior precisão. Segundo Cano et al. (2011), neste tipo de rede em que o consumo energético é uma grande restrição, o projeto da camada MAC torna-se um fator importante, uma vez que ela controla diretamente a operação do transceptor, que é o componente que mais consome energia em um nó sensor. Portanto, a camada MAC deve priorizar a redução do consumo de energia dos nós sensores. As diferentes fontes de desperdício de energia podem ser classificadas da seguinte forma:

Escuta ociosa: A escuta ociosa (*Idle Listening*) ocorre quando um nó escuta o canal para uma possível recepção, mas não há transmissões em curso. Escuta ociosa tem sido identificada como a principal fonte de desperdício de energia em um nó sensor, principalmente devido às baixas taxas de tráfego comumente encontradas em RSSFs.

Colisões Quando dois (ou mais) nós transmitem simultaneamente, o receptor pode ser incapaz de decodificar qualquer uma das mensagens envolvidas. Este problema implica que os nós emissores desperdiçam energia transmitindo e os nós receptores desperdiçam energia recebendo dados que serão descartados.

Overhearing: Esta fonte de dissipação de energia ocorre quando um sensor desperdiça energia ao receber uma mensagem que é destinado a um outro nó.

Overhead: As mensagens de dados em redes de sensores são geralmente pequenas; portanto, cabeçalhos e outros tipos de informações de controle implicam em desperdício de energia para RSSF.

Protocolos MAC projetados para redes sem fio locais (WLANs) ou redes de malha, como a função de coordenação distribuída (DCF - *Distributed Coordination Function*) definida no padrão IEEE 802.11 (IEEE STANDARDS ASSOCIATION, 2012), não são adequados para as necessidades especiais das RSSF. Eles não consideram as diferentes restrições das RSSF. Normalmente, esses protocolos assumem que nós são dispositivos com memória elevada e hardware preciso de sincronização e, sobretudo, com recursos energéticos ilimitados. Conseqüentemente, os protocolos não abordam as fontes de desperdício de energia descritas anteriormente.

O comportamento comum dos protocolos MAC para o padrão IEEE 802.11, é ouvir o canal continuamente para uma possível recepção. No entanto, em RSSF, este modelo irá consumir rapidamente os recursos de energia de um nó sensor, devido à escuta ociosa. Além disso, como os nós permanecem constantemente ouvindo o canal, o *overhearing* é comum nesse tipo de protocolo. O *overhead* também é considerável, uma vez que o comprimento das mensagens de dados das aplicações são significativamente mais elevados do que os comprimentos encontrados em RSSFs. A única fonte de desperdício de energia que estes protocolos MAC consideram, são as colisões, que além do desperdício de energia, reduzem consideravelmente o desempenho da rede.

Os protocolos MAC tradicionais são comumente focados na taxa de transferência ou latência, enquanto que, em RSSF, essas métricas são renegociadas a favor da diminuição do consumo energético. Embora possa ocorrer aumentos esporádicos na carga da rede devido a, por exemplo, as detecções de eventos, RSSF geralmente funcionam com cargas baixas. Portanto, nem sempre é necessário proporcionar um elevado rendimento ou redução de atraso para a aplicação.

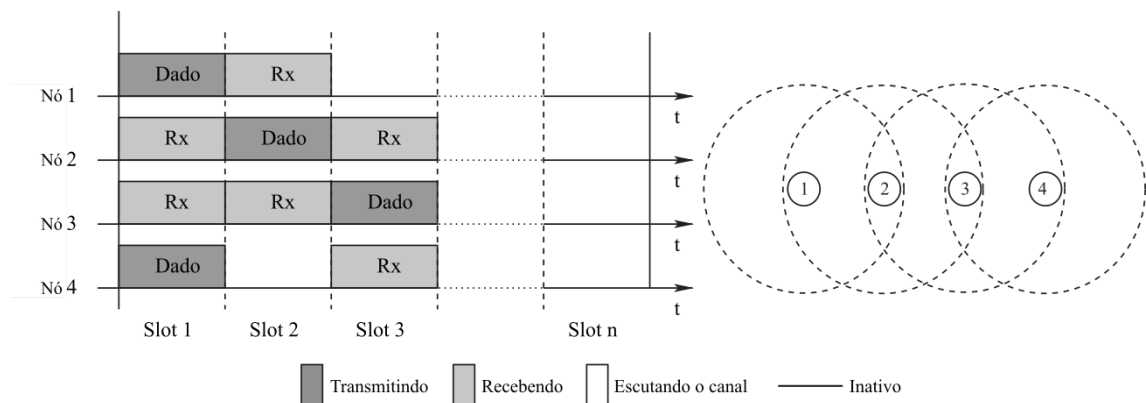
Tipos de Protocolos da Camada MAC para RSSF

Existe um grande número de protocolos MAC especialmente concebidos para RSSF. Para economizar energia, especialmente a energia desperdiçada por causa da escuta ociosa, a abordagem mais comum é colocar o transceptor em modo inativo pelo máximo de tempo possível, já que o modo inativo consome substancialmente menos energia do que os outros modos disponíveis como: ocioso, transmitindo ou recebendo mensagem. No modo inativo, um nó sensor não é capaz de receber ou transmitir mensagens para o meio. Esta solução reduz significativamente o desperdício de energia de escuta ociosa. No entanto, mecanismos específicos

devem ser definidos para assegurar que um nó sensor entre em modo ativo para receber ou retransmitir mensagens de seus nós vizinhos. Segundo Cano et al. (2011), existem três categorias gerais para abordar esta questão:

Protocolos baseados em TDMA (*Time Division Multiple Access*): nesta categoria de protocolos, o tempo é dividido em quadros (*frames*) e cada quadro é dividido em segmentos (*slots*). Para cada nó, são atribuídos segmentos específicos no quadro para que possa transmitir ou receber mensagens. Os nós entram em atividade somente nestes segmentos, e durante o resto do tempo, permanecem inativos. Para comunicar com seus vizinhos, o nó deve saber em quais segmentos eles estarão em atividade. A fim de evitar colisões, os nós em até dois saltos de distância, devem ter períodos para a transmissão diferentes entre si. Uma representação esquemática é apresentada na Figura 2.3.

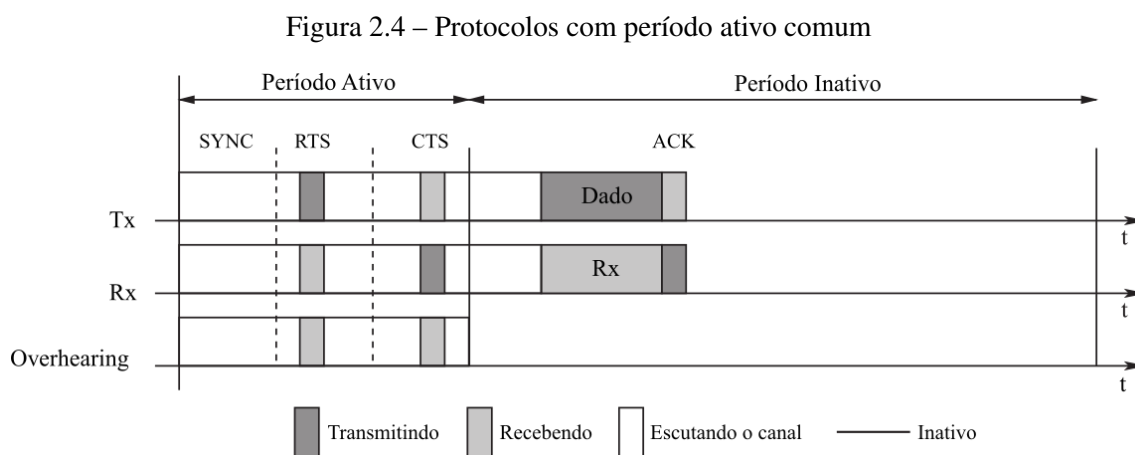
Figura 2.3 – Protocolos baseados em TDMA



Fonte: (CANO et al., 2011)

Os nós 2 e 3 não estão a mais de dois saltos de distância do nó 1, portanto seus períodos de transmissão não podem ser iguais. Já para o nó 4, não há restrições de períodos em relação ao nó 1. No segmento de tempo em que o nó 1 está transmitindo, seus vizinhos estão no período de recepção. O mesmo se aplicará para os demais nós, de forma que todos possam trocar mensagens entre si sem ocorrência de colisões. Este tipo de protocolo provê atrasos controlados, taxas altas de transferência e redução ou eliminação de colisões, mas requer uma sincronização rigorosa entre os nós sensores. Além disso, há sempre um custo de *overhead* associado com a criação e manutenção da atribuição dos segmentos de tempo, e custos para manter a sincronização entre os nós. Conforme Cano et al. (2011), os protocolos PEDAMACS (ERGEN; VARAIYA, 2006) e LMAC (HOESEL; HAVINGA, 2004) são alguns exemplos que representam esta categoria.

Protocolos com período ativo comum: estes protocolos sincronizam os nós sensores para que os ciclos de atividade/inatividade ocorram ao mesmo tempo. A Figura 2.4 representa a funcionalidade de um dos protocolos mais conhecidos nesta categoria: o protocolo S-MAC (YE; HEIDEMANN; ESTRIN, 2002). Conforme ilustrado na figura, os nós sensores prontos para transmitir esperam até o período ativo. Os autores do S-MAC definem que um *handshake* (utilizando RTS/CTS) é feito no período ativo, mas a comunicação de dados é feita no período inativo, permitindo que os nós em *overhearing* possam dormir. Apesar que um certo grau de sincronização também seja necessário nestes protocolos, este requisito não é tão rigoroso como em protocolos baseados em TDMA. Entretanto, também possuem um custo associado com a criação e manutenção da escala de horários. Segundo Cano et al. (2011), o protocolo T-MAC (DAM; LANGENDOEN, 2003) e SCP-MAC (YE; SILVA; HEIDEMANN, 2006) também são exemplos que representam esta categoria.

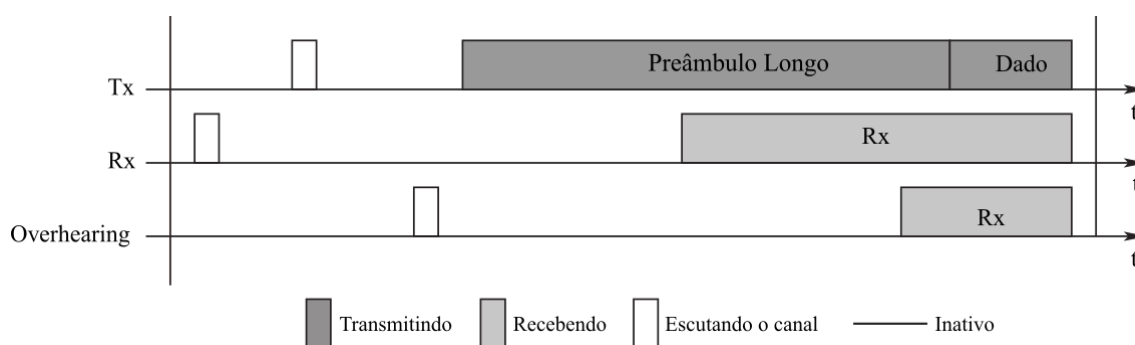


Fonte: (CANO et al., 2011)

Protocolos assíncronos: nesta categoria de protocolo, não é necessária sincronização, uma vez que cada nó possui ciclos de atividade/inatividade independentemente de outros nós na rede. Estes protocolos implementam diferentes mecanismos para assegurar que um nó estará ativo para receber ou retransmitir as mensagens de seus vizinhos. Estes protocolos podem ser divididos em duas categorias: amostragem de preâmbulo (*preamble sampling*) e comunicação iniciada pelo receptor. Na amostragem de preâmbulo, os nós sensores entram em atividade apenas para verificar se o canal está ocupado. Se for detectada alguma transmissão, o nó permanece ativo, caso contrário, volta para o estado inativo. Um nó sensor que possui mensagem para transmitir, primeiro envia um longo preâmbulo para sobrepôr o intervalo de amostragem do canal pelo receptor. O nó receptor irá manter-se

ativo até o fim do preâmbulo para então receber a mensagem de dados. Este comportamento está representado na Figura 2.5. Pode-se observar que o preâmbulo longo faz com que o receptor e os nós em *overhearing* fiquem ativos para receber a mensagem. Por outro lado, na categoria de protocolos iniciados pelo receptor, nós sensores enviam um anúncio quando entram em atividade. Então, um nó sensor com uma mensagem para transmitir espera até receber o anúncio enviado pelo nó receptor. Os protocolos assíncronos não possuem um custo associado com a criação e manutenção da escala de horários, mas possuem um custo associado ao *overhearing* e, em sua maioria, não possuem mecanismos para combater colisões. Conforme Cano et al. (2011), os protocolos B-MAC (POLASTRE; HILL; CULLER, 2004), X-MAC (BUETTNER et al., 2006) e WiseMAC (EL-HOIYDI; DECOTIGNIE, 2004) são alguns exemplos de protocolos assíncronos que utilizam a amostragem de preâmbulo. Como exemplo de protocolo em que a comunicação é iniciada pelo receptor, pode ser citado o RI-MAC (SUN; GUREWITZ; JOHNSON, 2008).

Figura 2.5 – Protocolo assíncrono com amostragem de preâmbulo



Fonte: (CANO et al., 2011)

Segundo Cano et al. (2011), entre os diferentes protocolos MAC projetados para RSSF, abordagens assíncronas e amostragem de preâmbulo fornecem algumas características que os tornam atraentes para RSSF. Comparada com protocolos de período ativo comum, a amostragem de preâmbulo fornece menos complexidade e custo, já que nenhuma escala de horário tem que ser criada, notificada e mantida. Além disso, a amostragem de preâmbulo consome menos energia em comparação aos protocolos de período ativo comum quando a carga de tráfego é baixa. Esta diferença é devido ao curto tempo de amostragem do canal, o que permite a um nó sensor retornar imediatamente para o estado inativo se o meio é encontrado ocioso. Entretanto, nos protocolos de período ativo comum, o período ativo é geralmente grande, uma vez que deve

ser suficiente para a transmissão de mensagens de sincronização, controle e dados. Se comparado com os protocolos MAC baseados em TDMA, a amostragem de preâmbulo é muito menos complexa, pois nenhuma escala de horário deve ser computada e distribuída entre os nós.

Conforme (LI; LI; LIU, 2014), uma vez que os protocolos assíncronos não possuem restrições decorrente da sincronização, possibilitam a criação de RSSF robustas em ambientes dinâmicos, sendo assim vantajosos para implantações em larga escala. Entretanto, o uso de preâmbulos longos como no protocolo B-MAC, também ocasionam um maior nível de *overhead* e *overhearing* na rede.

Mantendo os benefícios dos protocolos assíncronos mas buscando menores gastos energéticos, diferentes abordagens são propostas para reduzir o *overhead* e *overhearing* nos protocolos assíncronos. Conforme Cano et al. (2011), estas abordagens podem ser divididas em três categorias: *i*) divisão do preâmbulo longo em sequências de preâmbulos menores, *ii*) obtenção da informação dos horários de atividade dos nós, e *iii*) adaptação da duração dos ciclos de trabalho.

Divisão do preâmbulo: esta categoria de protocolos divide o preâmbulo longo numa série de preâmbulos curtos com o principal objetivo de incluir algumas informações úteis nestas mensagens. A abordagem mais comum é incluir o identificador do destino (ID) nos preâmbulos curtos. Isto permite que os nós em *overhearing* retornem ao modo inativo, reduzindo o custo de *overhearing* (como em ENBMAC (LIM et al., 2006)). Este tipo de protocolo é representado na Figura 2.6 (exemplo 2). É possível observar que quando a informação do destino é incluída, o nó em *overhearing* pode voltar imediatamente para o modo inativo.

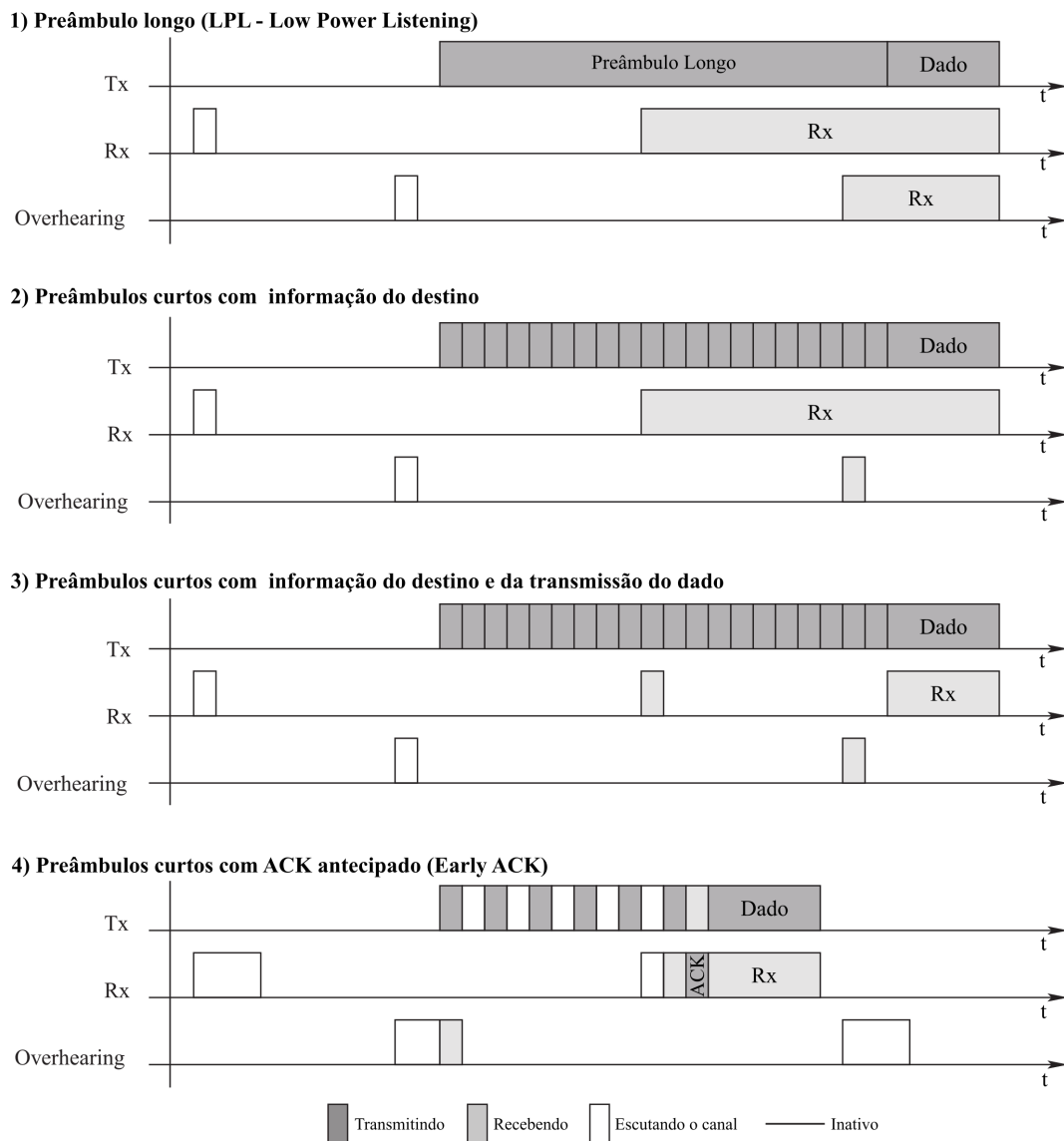
Outras abordagens também incluem informações sobre quando a transmissão dos dados se iniciará. Desta forma, o destino pode ficar inativo após receber o preâmbulo, e entrar em atividade somente no momento de receber os dados (como em BMAC+ (AVVENUTI et al., 2006) e SpeckMAC-B (WONG; ARVIND, 2006)). A Figura 2.6 (exemplo 3) apresenta esta abordagem.

Existem abordagens que enviam um ACK antecipado (eACK - *early* ACK) que interrompe a transmissão de preâmbulos. Nesta abordagem, entre cada preâmbulo enviado, o nó emissor mantém um intervalo para realizar a escuta do canal. Quando o nó de destino receber um preâmbulo, imediatamente enviará uma mensagem eACK. Assim que o emissor detectar o eACK, o envio de preâmbulos será cessado e a transmissão da mensa-

gem de dados será iniciada. Esta abordagem exige a transmissão de um número menor de preâmbulos, além de permitir que a mensagem de dados seja enviada mais cedo, reduzindo a latência (como em X-MAC (BUETTNER et al., 2006), CMAC (LIU; FAN; SINHA, 2009) e TICER (LIN; RABAEY; WOLISZ, 2004)). A Figura 2.6 (exemplo 4), ilustra o funcionamento dessa classe de protocolos.

Também há outras abordagens que ao invés de enviar rajadas de preâmbulos curtos, enviam repetições da mensagem de dados. Desta forma, o destino pode imediatamente receber a mensagem de dados quando entrar em atividade (como em MX-MAC (MERLIN; HEINZELMAN, 2007)).

Figura 2.6 – Representação esquemática de tipos de amostragem de preâmbulos



Fonte: (CANO et al., 2011)

Informação dos horários de atividade de outros nós: nesta classe de protocolos os nós mantêm a informação de quando os nós receptores entrarão em atividade. Dessa forma, podem reduzir o tamanho do preâmbulo longo. O tamanho do preâmbulo pode ser apenas o suficiente para considerar o desvio de relógio dos nós. Portanto, quanto mais recente a informação do tempo de atividade do receptor, menor será o preâmbulo. As informações de quando o nó entrará em atividade são informadas nas mensagens ACK. Esta técnica apareceu pela primeira vez em WiseMAC (EL-HOIYDI; DECOTIGNIE, 2004).

Adaptação da duração dos ciclos de trabalho: Existem diferentes abordagens para adaptar o ciclo de trabalho e, também, a duração do preâmbulo para atingir um objetivo específico, geralmente para lidar com aumentos na carga de tráfego. Alguns protocolos adaptam o ciclo de trabalho com base em solicitações dos vizinhos. Por exemplo, em WiseMAC (EL-HOIYDI; DECOTIGNIE, 2004), o emissor utiliza um *bit* no cabeçalho da mensagem de dados para indicar que o receptor deve permanecer ativo (não seguir o ciclo de trabalho), quando há mais mensagens para serem enviadas.

Uma abordagem semelhante é definida em BEAM (ANWANDER et al., 2010), o emissor define um *bit* no cabeçalho da mensagem de dados se há mais mensagens a serem enviadas. Se o *bit* é marcado, o destino dobra seu ciclo de trabalho, permitindo que o emissor reduza o preâmbulo longo, aumentando a vazão. Outro exemplo é o protocolo AS-MAC (LIU; ZHANG; ZHANG, 2007). Em AS-MAC, os transmissores incluem o tempo da próxima transmissão nas mensagens, de modo que o receptor possa ajustar o seu ciclo de trabalho para a recepção. Se o emissor indicar que não há mais mensagens a transmitir, o receptor aumenta seu período de inatividade.

Outra técnica adapta o ciclo de trabalho baseado na carga de tráfego observada. Um exemplo é o trabalho apresentado em Avvenuti e Vecchio (2008), que propõe uma extensão do B-MAC+ e define que, após detectar um aumento no número de mensagens recebidas, o nó deve aumentar seu ciclo de trabalho. A configuração do novo ciclo de trabalho é informada aos vizinhos nas rajadas de preâmbulos que antecedem o envio da mensagem de dados. Uma abordagem semelhante é o protocolo BoostMAC (STONE; COLAGROSSO, 2007). Nesta abordagem, os nós sensores aumentam seu período inativo em etapas aditivas (até que um certo valor seja alcançado) se encontrarem o canal livre quando acordam. Caso contrário, se um alto nível de tráfego é observado, eles diminuem o tempo inativo de forma multiplicativa. Em vez de anunciar o período de inatividade que cada nó está

usando, os emissores decidem qual comprimento de preâmbulo será usado com base em seu próprio tráfego.

Outro exemplo é o protocolo MaxMAC (HURNI; BRAUN, 2010), que fornece uma extensão para o WiseMAC. Usando o MaxMAC, os nós dobram seu ciclo de trabalho se o número de mensagens recebidas exceder um limiar. MaxMAC duplica o ciclo de trabalho mais uma vez se um segundo limiar é ultrapassado. Finalmente, depois de ultrapassar um terceiro limiar, os nós sensores começam a trabalhar de uma forma CSMA sem entrar em modo inativo. As mensagens ACK são usadas para informar sobre a mudança no ciclo de trabalho e a duração da nova configuração. Portanto, baseado nestas informações, os nós vizinhos podem adaptar o comprimento da rajada de preâmbulos curtos.

Existem outras abordagens que adaptam o ciclo de trabalho com base na posição do nó na topologia. Por exemplo, EA-ALPL (JURDAK; BALDI; LOPES, 2005) assume uma topologia em árvore e permite que cada nó sensor estabeleça seu próprio ciclo de trabalho de acordo com sua carga e seu número de descendentes na árvore de roteamento. Os nós descobrem o número de filhos contando o número de mensagens que enviaram em um certo intervalo. As informações do ciclo de trabalho são trocadas entre vizinhos e armazenadas. Da mesma forma, em (SANCHEZ; CHAUDET; REBAUDENGO, 2009), cada nó seleciona seu ciclo de trabalho dependendo de sua posição na árvore. Os nós que estão mais próximos da estação base definem um pequeno tempo de inatividade, enquanto os nós nas folhas configuram um tempo de inatividade mais longo.

2.5 Camada de Rede

Na maioria das aplicações, os nós sensores possuem limitações na fonte de energia e na largura de banda de comunicação. Sendo assim, a necessidade da consciência energética em todas as camadas da pilha de protocolos é importante. Tais limitações, combinadas com um grande número de nós sensores, apresentam muitos desafios para a concepção e gestão das RSSF. Por exemplo, na camada de rede, é necessário encontrar métodos energeticamente eficientes para a descoberta de rotas e encaminhamento de dados dos nós sensores para a estação base, de modo que o tempo de vida da rede seja maximizado.

Devido às características únicas das RSSF, os protocolos de roteamento desenvolvidos para redes cabeadas e outras redes sem fio, como redes móveis *Ad Hoc* (MANET - *mobile ad hoc network*), muitas vezes não são adequados para RSSF. Segundo (ABDULLAH; EHSAN,

2014), a maioria das redes de sensores são específicas para cada aplicação, portanto possuem diferentes requisitos. Dependendo da aplicação, diferentes arquiteturas, objetivos e restrições de projeto são consideradas para as redes de sensores sem fio. Conforme (YANG, 2013), as principais características dos protocolos de roteamento em RSSF são descritas a seguir:

- Um dos principais objetivos do protocolo de roteamento em RSSF é reduzir o consumo de energia, enquanto que protocolos de roteamento tradicionais são projetados para alcançar altos requisitos de qualidade de serviço (QoS), durante a transferência de dados.
- Os nós sensores são limitados em termos de energia, processamento e largura de banda. Portanto, exigem uma gestão cuidadosa de seus recursos.
- Construir um esquema de endereçamento global para a implantação de um grande número de nós, ocasiona *overhead* na manutenção dos números de identificação (ID). Assim, os protocolos tradicionais baseados em IP podem não ser adequados para redes de sensores sem fio.
- As redes de sensores são específicas para a aplicação, ou seja, os requisitos de projeto mudam conforme a aplicação.
- Os dados coletados pelos sensores na rede, são tipicamente baseados em fenômenos comuns, portanto há uma elevada probabilidade de que esses dados tenham alguma redundância que deva ser explorada para melhorar a utilização de energia e banda.
- Nós sensores que são implantados de forma *ad hoc* precisam da capacidade de auto-organização, especialmente pelo fato de que devem operar sem intervenções humanas.
- A maioria das aplicações em RSSF requerem que o encaminhamento dos dados sensoriais de múltiplos nós seja feito para uma determinada estação base. Isso, no entanto, não impede que o fluxo de dados ocorra de forma *multicast* ou ponto-a-ponto.
- A consciência da localização dos nós sensores é importante, uma vez que a coleta de dados é normalmente baseada na localização.
- Os nós sensores em RSSF são mais propensos a erros ou falhas, seja pelo *hardware* de baixo custo, pelo gerenciamento energético ou devido ao ambiente no qual se encontra. Portanto, o protocolo de roteamento deve funcionar efetivamente mesmo quando houver

falhas de nós na rede. Esta característica de tolerância a falhas, requer que o protocolo de roteamento tenha a capacidade de superar qualquer falha na rede, descobrindo e mantendo uma nova rota para transmitir os dados.

2.5.1 Tipos de Protocolos da Camada de Rede para RSSF

A tarefa de encontrar e manter rotas em RSSF não é trivial, as restrições de energia e mudanças bruscas na condição do nó, frequentemente causam alterações topológicas imprevisíveis. Para minimizar o consumo de energia, técnicas de roteamento para RSSF empregam algumas táticas de roteamento clássicas da literatura, além de táticas especiais como: agregação de dados, processamento em rede, *clustering*, atribuição de papéis diferentes para os nós e métodos centrados em dados.

De acordo com (ZIN et al., 2014), a classificação dos protocolos de roteamento em redes sensores é baseada em três aspectos: *i*) modo de funcionamento, *ii*) forma de participação dos nós, e *iii*) estrutura da rede.

Modo de funcionamento: de acordo com a forma de funcionamento, os protocolos de roteamento podem ser divididos em: proativo, reativo e híbrido.

Proativo: uma tabela de roteamento é gerada em cada nó, de modo que as informações de roteamento são mantidas para cada nó da rede. As informações de roteamento são disseminadas periodicamente, atualizando as informações mantidas pelos nós. Nesta categoria de roteamento, todas as rotas são computadas antes da necessidade de encaminhar alguma mensagem.

Reativo: nenhuma tabela de roteamento é gerada e a descoberta da rota é feita conforme necessário ou com base na demanda. As informações de rotas obtidas podem ser mantidas para referências futuras. Esta estratégia reduz a sobrecarga de comunicação causada por inundação (*flooding*) de mensagens de roteamento na rede.

Híbrido: combina as características de roteamento proativo e reativo. Uma das abordagens dos protocolos híbridos, é calcular primeiro todas as rotas e em seguida, atualizar e aprimorar as rotas no momento do roteamento.

Forma de participação dos nós: de acordo com o modo de participação dos nós, os protocolos podem ser divididos em: direto, plano e baseado em agrupamento (*clustering*).

Direto: permite que os nós enviem as mensagens diretamente para o destino final.

Plano: se algum nó precisa enviar alguma mensagem, primeiramente encontrará uma rota válida até o destino final, para então depois encaminhá-la.

Baseado em agrupamento: a rede é dividida em grupos (*clusters*), sendo que cada grupo possui um nó líder (*cluster head*), e apenas o nó líder irá comunicar diretamente com a estação de base. A criação de grupos e atribuição de tarefas especiais para formar os líderes podem contribuir para a escalabilidade, tempo de vida e eficiência energética da rede. O agrupamento é uma forma eficaz de reduzir o consumo de energia dentro de um grupo, realizando agregação e a fusão de dados, a fim de diminuir o número de mensagens transmitidas para a estação base.

Estrutura da rede: de acordo com a estrutura da rede, os protocolos podem ser divididos em: centrado em dados, hierárquico, baseado em localização ou ciente de qualidade de serviço (QoS - *Quality of Service*).

Centrado em dados: os protocolos centrados em dados são baseados em consultas e dependem da nomeação ou identificação dos dados, sendo responsáveis pela eliminação de transmissões de dados redundantes.

Hierárquico: usados para realizar roteamentos energeticamente eficientes, onde os nós com maior nível de energia podem ser usados para processar e encaminhar as informações, enquanto que os nós com menor nível de energia, usados para o sensoramento da área de interesse.

Baseados em localização: neste tipo de roteamento, os nós sensores são endereçados por meio de suas localizações. A distância entre os nós vizinhos pode ser estimada com base na intensidade dos sinais de rádio. As coordenadas relativas dos nós vizinhos podem ser obtidas por troca dessas informações entre os vizinhos. Alternativamente, a posição dos nós pode ser obtida diretamente através da comunicação com um satélite usando GPS, se os nós estiverem equipados com receptores GPS. Para economizar energia, alguns esquemas podem coordenar o período inativo dos nós baseado em suas localizações.

Cientes de QoS: o roteamento ciente de QoS concentra-se em vários requisitos da camada de rede na operação de roteamento, como latência e confiabilidade.

2.5.2 Tipos de Transmissão de Dados

Diferentes metodologias de transmissão de dados podem ser utilizados em uma rede de sensores sem fio. As principais metodologias são: *unicast*, *multicast*, *anycast* e *broadcast*, conforme ilustrado na Figura 2.7.

Unicast: tipo de encaminhamento no qual uma mensagem é endereçada a um nó específico.

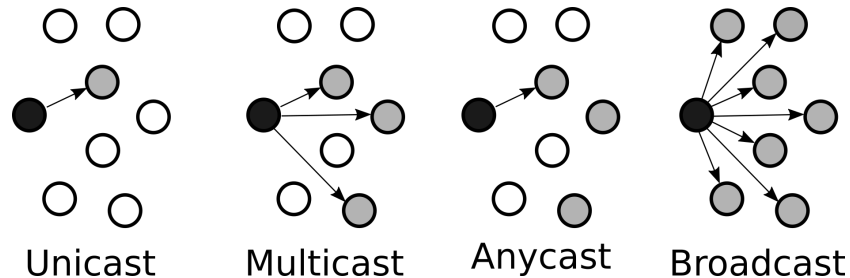
Na transmissão *unicast*, há apenas um remetente e um receptor.

Multicast: tipo de encaminhamento no qual uma mensagem é endereçada a um grupo específico de nós.

Anycast: tipo de encaminhamento no qual uma mensagem é endereçada a um grupo de nós, mas apenas um nó do grupo é eleito para o encaminhamento da mensagem.

Broadcast: tipo de encaminhamento no qual uma mensagem é endereçada a todos os nós na rede.

Figura 2.7 – Tipos de Transmissão de Dados



2.6 Abordagem *Cross-Layer*

O crescente desenvolvimento de novas tecnologias e dispositivos conectados em rede, resultou no aumento da complexidade e diversidade dos sistemas e, conseqüentemente, na necessidade de discussão arquitetural sobre como englobar essas novas tecnologias que possuem características e requisitos bem diferentes.

A separação das tarefas de uma rede em camadas consiste numa forma de dividir e organizar o *software*, reduzindo complexidade do sistema e aumentando sua compatibilidade. Contudo, nem todas atribuições podem ser modularizados em camadas. Gerenciamento energético, segurança e tolerância a falhas são exemplos de atribuições cujas implementações se espalham pelas diversas camadas da arquitetura.

O projeto arquitetural das redes de sensores sem fio é baseada no padrão IEEE 802.15.4 (IEEE STANDARDS ASSOCIATION, 2015). Nesse modelo, cada camada pode executar serviços para a camada imediatamente acima e fazer requisições para a camada imediatamente abaixo, ou seja, a comunicação ocorre apenas entre camadas adjacentes e por meio de interfaces conhecidas e bem definidas. Portanto, os protocolos pertencentes às diferentes camadas não podem acessar qualquer informação que não esteja definida nas regras de comunicação estabelecidas pela arquitetura.

A maioria dos protocolos de comunicação, explorando a natureza colaborativa das RSSF, melhoram a eficiência energética até certo ponto. A principal semelhança destes protocolos é que eles seguem as arquiteturas de protocolo em camadas tradicionais. A maioria destes protocolos de comunicação são desenvolvidos individualmente para cada camada da arquitetura.

Embora estes protocolos possam atingir um desempenho alto em termos das métricas relacionadas a cada camada individualmente, não são otimizados em conjunto para maximizar o desempenho global da rede e minimizar o gasto de energia. Considerando a escassa energia e os recursos de processamento das RSSF, projetistas de protocolos têm utilizado uma alternativa que é permitir a comunicação direta entre os protocolos de camadas não adjacentes, ou o compartilhamento de variáveis entre as camadas. Tal violação às regras de comunicação da arquitetura em camadas tem sido denominada arquitetura *cross-layer* (*cross-layer design*) (MENDES; RODRIGUES, 2011). A arquitetura *cross-layer* é essencialmente uma modificação na pilha de protocolos existente.

As configurações operacionais de uma dada camada não são feitas apenas localmente, mas considerando as necessidades de outras camadas, a fim de obter melhores resultados de otimização. Diferentes soluções utilizando abordagem *cross-layer* já foram propostas na literatura, demonstrando alcançar melhores ganhos de desempenho em relação às abordagens de camadas de forma tradicional (AKYILDIZ; VURAN; AKAN, 2006), (MENDES; RODRIGUES, 2011), (RAULT; BOUABDALLAH; CHALLAL, 2014), (TOKLU; ERDEM, 2014) e (VURAN; AKYILDIZ, 2010). Os objetivos comuns da abordagem *cross-layer* são a redução do consumo de energia, roteamento eficiente, prover QoS e a programação eficiente do quadro de horários.

2.7 Comunicação Anycast

A integração de funcionalidades da camada MAC e da camada de rede foram exploradas por vários projetos na última década (MENDES; RODRIGUES, 2011). Nesse sentido, uma abordagem *cross-layer* eficiente é a *anycast*, uma classe de protocolos que tira proveito da característica da redundância inerente das RSSF densas, para reduzir o atraso causado pelo ciclo ativo/inativo dos nós sensores (ASHRAF; VAIDYA; KRAVETS, 2011), (TAN; CHAN, 2010), (KIM et al., 2010) e (ZENG et al., 2007).

Os protocolos MAC baseados em *anycast* são geralmente parte de abordagens *cross-layer*, nas quais interações com outras camadas definem o comportamento de ambos. Abordagens *cross-layer* que exploram combinações das camadas MAC e de rede podem ser divididas em duas classes de comunicação. Na primeira classe, a estrutura tradicional em camadas é preservada, mas cada camada pode ser informada sobre as condições das outras. Assim, protocolos MAC exploram dados fornecidos pela camada de roteamento, de forma que as estruturas de cada camada ainda permanecem intactas. Na segunda classe, por outro lado, há muito a se ganhar repensando ambas camadas de forma unificada, concentrando funcionalidades de ambas (VURAN; AKYILDIZ, 2010). Os protocolos *anycast* existentes, normalmente unificam a funcionalidade de protocolos MAC assíncronos baseados em preâmbulos, como X-MAC (BUETTNER et al., 2006) e WiseMAC (EL-HOIYDI; DECOTIGNIE, 2004), com protocolos de roteamento geográficos (KIM et al., 2010) e (ZENG et al., 2007).

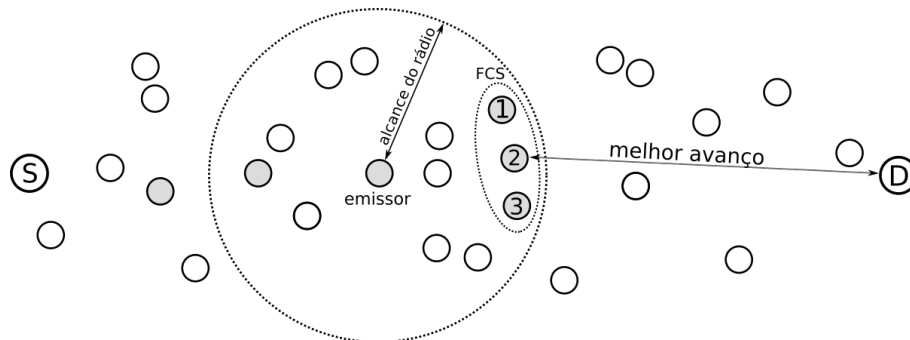
Protocolos baseados em comunicação *anycast*, aproveitam a redundância do caminho, isto é, vários nós candidatos disponíveis para realizar a retransmissão da mensagem, reduzindo a latência causada pelo ciclo de trabalho dos nós. Diferentemente dos protocolos de comunicação *unicast* tradicionais, não é escolhido um único nó para a retransmissão, mas sim, um subconjunto de nós vizinhos denominado FCS (*Forwarding Candidate Set*), com condições apropriadas para encaminhar a mensagem.

A decisão se um nó está incluído no FCS pode ser realizada no lado do emissor ou do receptor. Quando o emissor é responsável por selecionar o FCS, ele deve conhecer a posição geográfica e outras informações relevantes do estado dos nós vizinhos (como em (ASHRAF; VAIDYA; KRAVETS, 2011), (CHOUDHURY; VAIDYA, 2004), (TAN; CHAN, 2010), (ZENG et al., 2007)). Dessa forma, os membros do FCS são selecionados e seus identificadores são informados nos preâmbulos através da camada MAC. Quando a decisão é tomada no lado do receptor, o emissor transmite no preâmbulo os parâmetros necessários para o processo de deci-

são. Ao receber o preâmbulo, cada potencial retransmissor decide se é candidato a retransmitir a mensagem de dados, com base nas informações recebidas (como em (CHENG et al., 2012), (LIU; FAN; SINHA, 2009), (VURAN; AKYILDIZ, 2010), (ZORZI; RAO, 2003)).

No contexto do roteamento geográfico, um possível FCS é o conjunto composto por nós que trazem uma melhoria em termos de redução da distância até o destino. A Figura 2.8 ilustra este exemplo.

Figura 2.8 – Princípio dos protocolos *anycast*



Os membros do FCS são compostos pelos três nós que promovem um maior avanço até o destino (nós 1, 2 e 3). O membro do FCS que responder primeiro ao preâmbulo, assume a função de retransmissor e envia de volta uma mensagem confirmando a recepção do preâmbulo. Assim como proposto em Buettner et al. (2006), essa mensagem é chamada neste trabalho de *eACK* (*early acknowledgment*). Quando o nó emissor recebe um *eACK*, ele cessa o envio de preâmbulos e começa o envio da mensagem de dados. Na Figura 2.8, os nós 1, 2 e 3 foram incluídos no FCS, uma vez que todos proporcionam um avanço aceitável para a mensagem de dados em direção ao destino *D*. Neste caso, o emissor receberá um *eACK* em 25% do tempo de ciclo, em média, enquanto que isto aconteceria em 50% do tempo de ciclo se um protocolo *unicast* fosse usado (apenas um membro no FCS).

3 Trabalhos Relacionados

Entre as propostas que abordam as redes assíncronas, GeRaF (ZORZI; RAO, 2003) foi um protocolo pioneiro a usar comunicação *anycast*, visando reduzir a latência em protocolos assíncronos com sondagem de preâmbulos. Neste protocolo, a vizinhança do nó emissor é dividida em regiões. O nó emissor começa sondando a região mais próxima do destino utilizando uma mensagem de preâmbulo. Se houver um nó ativo nessa região, o primeiro nó que responde com um sinal eACK é selecionado como o melhor receptor. Caso contrário, o nó emissor procede sondando a segunda região mais próxima do destino. Esse processo é realizado até que o emissor detecte um receptor ativo na região atual, ou que o emissor tenha sondado todas as regiões e não tenha detectado vizinhos em atividade. Neste último caso, a comunicação irá falhar.

Uma melhoria do GeRaF é apresentada pelo CMAC (LIU; FAN; SINHA, 2009), que elimina a necessidade de enviar preâmbulos específicos para cada região, durante a busca de nós receptores. CMAC mantém a divisão das regiões de encaminhamento, assim como o GeRaF. Entretanto, cada região possui um intervalo de resposta ao preâmbulo, sendo o primeiro intervalo utilizado pelos nós posicionados na região mais próxima do destino. Os nós que estão na mesma região e ouvem um preâmbulo ao mesmo tempo, respondem em tempos aleatórios com um eACK. CMAC define um avanço mínimo exigido (r_0) como um critério para selecionar os membros do FCS. Os vizinhos do emissor que não fornecem um avanço para o destino maior que r_0 , não estão autorizados a enviar de volta um eACK. A seleção de r_0 depende da carga de tráfego e da densidade da rede. O comprimento da mensagem de dados não é considerado.

Baseando-se no GeRaF, o protocolo *cross-layer* AGA-MAC (HEIMFARTH; GIACOMIN; ARAUJO, 2015) segue o mesmo princípio do CMAC, impondo um limiar no avanço para o destino. No AGA-MAC, este limiar depende do comprimento da mensagem de dados. Se a mensagem de dados é curta, o limiar é reduzido, aumentando a cardinalidade do FCS, consequentemente, diminuindo o tempo de avanço. Se a mensagem de dados for longa, o limiar é ampliado, diminuindo a cardinalidade do FCS, consequentemente, reduzindo os saltos necessários para alcançar o destino. Este foi o primeiro trabalho abordando a influência do comprimento da mensagem de dados na latência de RSSF que utilizam abordagem *anycast*.

Uma proposta *cross-layer* chamada Any-MAC (ASHRAF; VAIDYA; KRAVETS, 2011) considera a camada de rede e os protocolos MAC existentes com pequenas modificações em ambos. Somente protocolos MAC assíncronos são considerados. Propostas Any-MAC baseadas

em sondagem utilizam protocolos MAC que executam a amostragem periódica do canal para identificar transmissões na vizinhança como o X-MAC (BUETTNER et al., 2006). Os protocolos assíncronos híbridos usam tanto a troca de escala de horários quanto a sondagem do canal. A redução de latência é obtida seguindo o mesmo princípio do roteamento oportunista usado por GeRaF (ZORZI; RAO, 2003). O FCS é determinado na camada de rede do emissor de acordo com o custo de roteamento calculado para cada vizinho. A camada de rede informa o FCS para a camada MAC. O primeiro membro FCS que responde ao sinal do emissor é eleito o nó do próximo salto. Um maior número de membros FCS resulta em um menor tempo médio para a mensagem de dados prosseguir o encaminhamento. No entanto, como efeito negativo, há um incremento na chance de colisões no envio de mensagens eACK quando dois ou mais membros do FCS acordam a tempo de ouvir e responder ao preâmbulo.

Outro protocolo *cross-layer* inspirado no GeRaF é o PAX-MAC (HEIMFARTH et al., 2016). Além disso, PAX-MAC também se baseia no protocolo síncrono RMAC (DU; SAHA; JOHNSON, 2007), entretanto, adaptando suas características para protocolos assíncronos *any-cast*. RMAC é um protocolo onde os nós possuem os mesmos períodos de atividade, assim como S-MAC (YE; HEIDEMANN; ESTRIN, 2002). Diferentemente do S-MAC, o RMAC envia as mensagens de dados separadamente das mensagens de controle. Durante o período ativo as mensagens de controle, denominadas PION, são enviadas a vários saltos à frente. Desta forma, durante o período inativo, o canal de comunicação é reservado para transmissão de dados. No período inativo, apenas os nós que participarão da comunicação, ficarão ativos no horário programado para receber e enviar a mensagem de dados. PAX-MAC adota o mesmo conceito, entretanto, por ser um protocolo assíncrono, os nós não possuem períodos de atividade em comum. Portanto, envia preâmbulos à frente encontrando nós que irão participar da transmissão de dados, informando o tempo em que cada nó deverá entrar em atividade para recebê-lo. Os resultados alcançados pelos autores, demonstrou que o protocolo obteve menor latência em relação aos protocolos testados.

Apesar do protocolo ter obtido o melhor tempo na entrega de mensagens, possui algumas características que limitam seu desempenho na redução da latência. Para evitar uma colisão quando a mensagem de dados se aproxima dos preâmbulos que estão à frente, PAX-MAC suspende o envio de preâmbulos até a chegada dos dados. Esta estratégia evita uma colisão iminente entre preâmbulo e a mensagem de dados, mas prejudica o ganho na latência, uma vez que avanço dos preâmbulos é temporariamente interrompido. Para minimizar este problema, o

nó de origem mantém um atraso no envio da mensagem de dados, reduzindo as possibilidades de aproximação entre dados e preâmbulo. Entretanto, se não houver uma redução na velocidade média no avanço dos preâmbulos, que justifique o tempo definido para esse atraso, ocorrerá o aumento na latência total para a entrega da mensagem de dados.

Embora as estratégias adotadas em Heimfarth et al. (2016) sejam interessantes, ainda possuem deficiências que podem ser exploradas a fim de obter melhores resultados. Portanto, ainda cabe novos estudos que possam elaborar estratégias ainda mais eficientes.

3.1 PAX-MAC

Esta seção apresenta o protocolo no qual este trabalho se inspirou. PAX-MAC (HEIMFARTH et al., 2016) é um protocolo MAC *cross-layer* que integra funções da camada de rede e da camada MAC. O protocolo utiliza comunicação *anycast* associado ao roteamento geográfico, com o objetivo de reduzir a alta latência comum em protocolos MAC assíncronos.

Com a finalidade de poupar energia, os nós sensores operam em ciclos de trabalho (hora está ativo, hora inativo) sem sincronização com outros nós na rede. Quando um nó tem uma mensagem para transmitir, envia antes uma série de preâmbulos curtos para encontrar o nó do próximo salto. Entretanto, este modo de operação causa latência na rede, uma vez que o próximo nó transmissor irá receber o preâmbulo somente quando estiver em seu período ativo.

Para reduzir o tempo no estabelecimento da comunicação entre o nó transmissor e o nó do próximo salto, PAX-MAC emprega a comunicação *anycast*. Diferente da comunicação *unicast* no qual o nó transmissor determina somente um vizinho para receber a mensagem, a comunicação *anycast* define um grupo de vizinhos habilitados a receber a mensagem. O primeiro nó que entrar em atividade deste grupo, envia um eACK para o transmissor interrompendo a sequência de preâmbulos.

Diferente de outros protocolos *anycast*, PAX-MAC não envia os dados logo após receber uma mensagem eACK. O preâmbulo propaga avançando pelos nós transmissores de modo a criar uma reserva de tempo para a recepção da mensagem de dados que virá. A transmissão de dados é iniciada somente após um intervalo predefinido depois que o nó de origem (nó fonte) envia o primeiro preâmbulo. A ideia é enviar os preâmbulos na frente comunicando com os nós transmissores e reservando o canal de comunicação para o momento em que as mensagens de dados serão enviadas. Os preâmbulos carregam a informação do tempo em que os dados serão transmitidos para o nó do próximo salto. A cada salto este tempo é incrementado

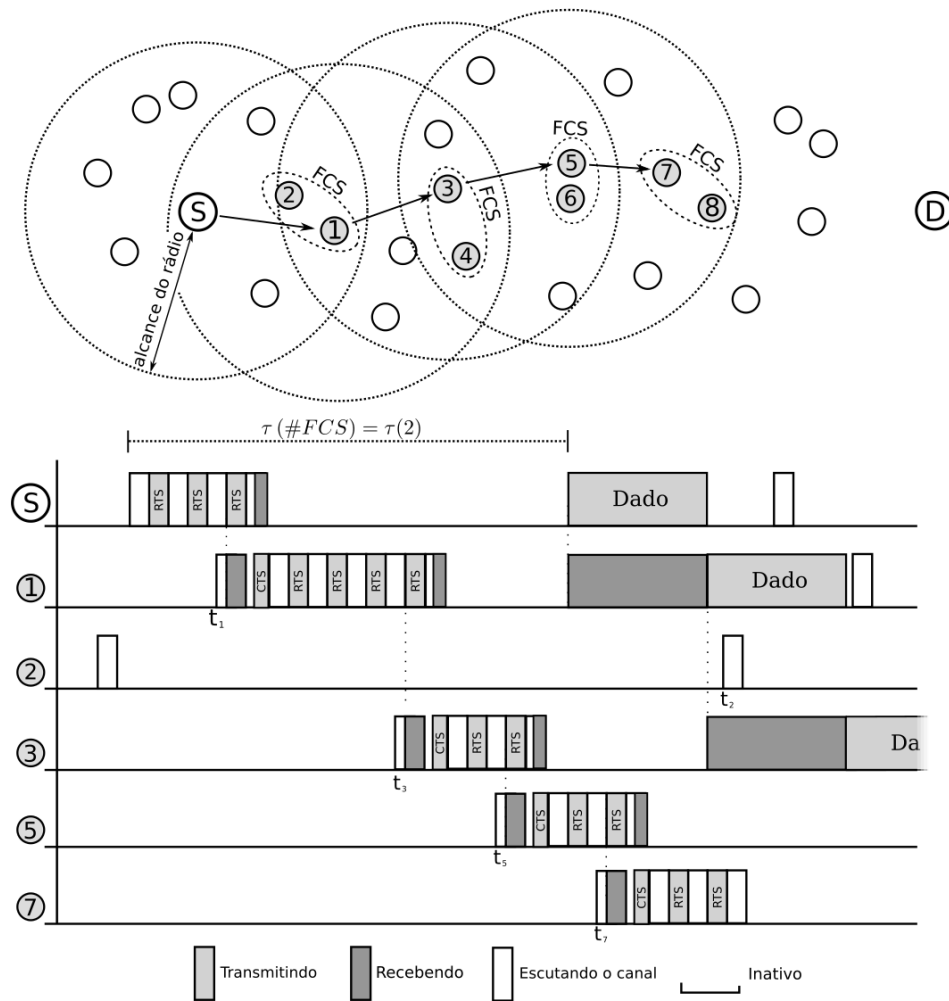
pelo tempo necessário para transmitir uma mensagem de dados. Dessa forma, os preâmbulos escalonam o tempo em que os nós transmissores entrarão em atividade para receber os dados. Ao mesmo tempo em que os dados são transmitidos, os preâmbulos estão alguns saltos adiante programando os nós dos próximos saltos.

A diferença de tempo entre a mensagem de dados e a série de preâmbulos é coordenada pelo nó de origem, mantendo um atraso inicial antes de enviar os dados. Os preâmbulos não se propagam a uma velocidade constante, portanto este tempo deve ser grande o suficiente para evitar colisões entre dados e preâmbulo em certo nó no caminho, mas não deve ser muito grande, caso contrário, isto aumentaria a latência total.

Demonstrando o funcionamento do PAX-MAC, a Figura 3.1 apresenta um exemplo de transmissão de mensagens. Neste exemplo, a origem S inicia uma transmissão endereçada ao destino D . A figura representa a transmissão nos quatro primeiros saltos. Para cada salto, o FCS é apresentado. Antes do envio da mensagem de dados, ocorre a transmissão de uma série de preâmbulos pelo nó de origem, intercalados com períodos de escuta para recepção do eACK. Como o nó 1, do FCS, entra em atividade primeiro, ele responde com uma mensagem eACK e assume o papel de próximo transmissor. Imediatamente, o nó 1 inicia uma série de preâmbulos para sinalizar sua intenção de transmitir a um nó de seu FCS, formado pelos nós 3 e 4.

No exemplo, o nó 3 do FCS é selecionado como próximo transmissor, uma vez que ele entra em atividade mais cedo e responde o preâmbulo do nó 1 com um eACK. O nó 1 é então programado para entrar no modo inativo, e torna-se ativo somente momento antes do nó de origem S começar a enviar a mensagem de dados. O processo é repetido, ou seja, o nó 3 começa imediatamente a transmitir preâmbulos para alcançar um nó pertencente ao seu FCS (formado pelos nós {5, 6}), programando para que o nó escolhido entre em atividade para receber a mensagem de dados que virá. O nó 5 entra em atividade primeiro e é selecionado devido à sua resposta eACK. Seguindo o mesmo processo, o nó 5 emite sua própria série de preâmbulos para sinalizar os membros de seu FCS. Durante a transmissão dos preâmbulos do nó 5, o nó S realiza, ao mesmo tempo, a transmissão da mensagem de dados para o nó 1, que foi programado para recebê-la. A partir desse momento, ocorre uma transmissão simultânea de dados e preâmbulos. Essas comunicações não interferem entre si, em vista da distância mantida entre as duas transmissões (maior que duas vezes o alcance dos rádios). O processo de envio de preâmbulos à frente enquanto a mensagem de dados segue alguns saltos atrás, continua até alcançar o nó de destino D .

Figura 3.1 – Visão geral do protocolo PAX-MAC



Fonte: (HEIMFARTH et al., 2016)

3.1.1 Seleção do FCS

Qualquer nó que pode promover algum avanço em direção ao destino poderia servir como nó transmissor, fazendo assim parte do caminho de transmissão da mensagem de dados. No entanto, a seleção somente dos nós que podem fornecer os maiores avanços, reduzirá o número total de saltos para alcançar o destino. Conseqüentemente, será utilizado menos energia nesta transmissão de dados. Portanto, o nó incluirá em seu FCS apenas os nós capazes de promover os maiores avanços na direção do destino. Isso é possível devido ao fato dos nós sensores terem conhecimento de suas próprias posições geográficas, além das posições geográficas dos nós vizinhos e do destino.

A decisão de quais nós que irão pertencer ao FCS é tomada no lado do nó emissor. Esta informação é incluída nos preâmbulos que serão enviados. Somente os membros do FCS são habilitados a responder um eACK em resposta ao preâmbulo.

O tempo de transmissão da mensagem de dados e o tempo médio necessário para encontrar um nó ativo no FCS são os mesmos. Isto é obtido regulando a cardinalidade do FCS com base no tamanho da mensagem. Assim, a mensagem de dados e os preâmbulos são transmitidos à mesma velocidade média, ao mesmo tempo, em locais diferentes da rede. O número de membros do FCS determina o número médio de preâmbulos necessários para identificar o nó do próximo salto. No PAX-MAC, o tempo para transmitir esses preâmbulos precisa ser menor ou igual ao tempo de transmissão da mensagem de dados. O número médio de preâmbulos necessários até encontrar o próximo transmissor em uma certa cardinalidade do FCS, é obtido pela Equação 3.1 conforme em Heimfarth et al. (2016):

$$p = \sum_{i=1}^{N_p} \left(\frac{i}{N_p} \right)^v \quad (3.1)$$

Em que v é o número de nós candidatos ao encaminhamento (FCS), p é a média de preâmbulos necessária até que o primeiro nó do FCS acorde e N_p é o número máximo de preâmbulos que se consegue enviar durante o período de inatividade estabelecido para os nós. A média de preâmbulo enviados até contactar o nó do próximo salto, determina a velocidade média do avanço dos preâmbulos. Uma vez que o tamanho da mensagem de dados é conhecida, a velocidade de avanço também pode ser calculada. Desse modo, é possível determinar o tempo τ , que é o atraso para liberar a mensagem de dados após o início do primeiro preâmbulo no nó de origem.

3.1.2 Colisão Iminente

As mensagens de dados são sempre enviadas com um atraso em relação aos preâmbulos. Apesar desta precaução, o tempo de propagação do preâmbulo pode ser mais longo do que o esperado, resultando em colisões. No exemplo representado na Figura 3.1, se o número total de preâmbulos necessários para fazer contato com o nó 5, leva um tempo maior que esperado por S para liberar a mensagem de dados, uma colisão correrá no nó 1. Da mesma forma, se o nó 5 estiver enviando preâmbulos para seu FCS e o nó 1 começar a enviar a mensagem de dados para o nó 3, ocorrerá uma colisão no nó 3. Esta colisão é especialmente prejudicial, pois irá desencadear uma outra transmissão de dados, desperdiçando tempo e energia.

O PAX-MAC propõe mecanismos para identificar a iminência de uma colisão e evitá-la. Quando uma colisão iminente é identificada, isto é, o nó transmissor percebe que o próximo preâmbulo que irá enviar está prestes a colidir com a mensagem de dados que está chegando

ao salto anterior, suspende a transmissão do preâmbulo. O nó transmissor atual aguardará a recepção dos dados e só então reiniciará as transmissões de preâmbulos, agindo portanto como o nó de origem (o primeiro nó transmissor). Isto é possível, pois quando o nó envia o preâmbulo para contactar o nó do próximo salto, está ciente da programação dos nós anteriores. Como sabe daqui quanto tempo os dados chegarão, consegue calcular em qual salto a mensagem de dados está naquele momento.

Entretanto, o mecanismo para evitar a colisão iminente possui um custo na latência total. O nó que está transmitindo os preâmbulos, têm que interromper estas transmissões até a recepção da mensagem de dados e então reiniciar o processo de envio da mensagem de dados, aumentando assim a latência total.

Uma estratégia para reduzir ou evitar possíveis situações de colisão iminente é aumentar o tempo τ , que é o atraso para liberar a mensagem de dados após o tempo de início do preâmbulo no nó de origem.

Quanto menor o tempo de espera τ no nó de origem, menor é a latência. No entanto, menores tempos de espera aumentam a chance de um evento de colisão iminente. Como consequência, a sobrecarga de reiniciar o processo de transmissão aumenta a latência. Portanto τ deve ser grande o suficiente para evitar colisões iminentes, mas não tão grande a ponto de prejudicar a latência total.

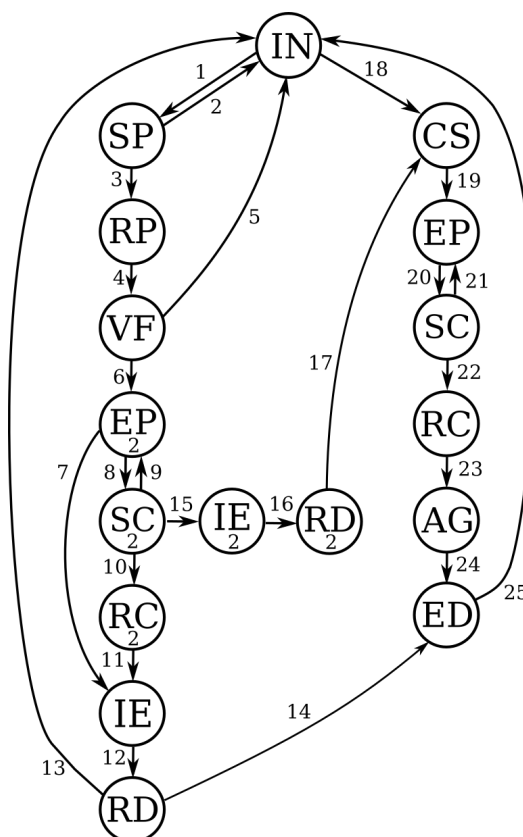
Seguindo o exemplo anterior, considerando que o nó 5 está enviando preâmbulos quando 1 está prestes a começar a enviar os dados para o nó 3. Nesse caso, o nó 5 cessa as transmissões e espera até receber a mensagem de dado do nó 3. Após receber os dados, 5 começa a enviar preâmbulos e aguarda o tempo de liberação dos dados com base no τ , da mesma forma que o nó de origem S .

3.1.3 Máquina de estados

Esta seção descreve a máquina de estados do protocolo PAX-MAC ilustrado na Figura 3.2. Os Quadros 3.1 e 3.2 apresentam os significados dos estados e suas transições respectivamente.

Durante o ciclo ativo/inativo, cada nó alterna entre os estados IN (inativo) e SP (sondando o canal em busca de preâmbulos) através das transições 1 e 2. O ciclo de trabalho determina o tempo de sondagem do canal em relação ao tempo total do ciclo.

Figura 3.2 – Máquina de estados do protocolo PAX-MAC



Quadro 3.1 – Lista dos estados da máquina de estados

Estado	Descrição
IN - Inativo	Nó está inoperante
SP - Sondando preâmbulos	Escutando o canal em busca de preâmbulos
RP - Recebendo preâmbulo	Preâmbulo está sendo recebido
VF - Verificando FCS	Verificando se o nó é um membro do FCS
EP - Enviando preâmbulo	Preâmbulo está sendo enviado
SC - Sondando confirmação	Escutando o canal em busca da mensagem de confirmação
RC - Recebendo confirmação	A mensagem de confirmação está sendo recebida
IE - Inativo aguardando dados	Nó fica inoperante até o tempo de receber os dados
RD - Recebendo dados	A mensagem de dados está sendo recebida
CS - <i>Carrier sensing</i>	Escutando o canal em busca de atividade
AG - Aguardando τ	Aguardando o período predefinido antes de enviar o dados
ED - Enviando dados	Enviando a mensagem de dados

Após receber a mensagem vinda das camadas superiores, é iniciada uma nova transmissão, direcionando ao estado CS (*carrier sensing*) através da transição 18. Se o canal estiver livre, a transição 19 mudará o estado para EP (enviando preâmbulo). Neste estado, o preâmbulo é transmitido em *broadcast*. Pela transição 20, o estado é alterado para SC (sondando a mensagem de confirmação). Neste estado, o protocolo MAC aguarda uma resposta de um nó vizinho

Quadro 3.2 – Lista das transições dos estados da máquina de estados

Transição	Descrição
1	Fim do tempo no modo inativo
2	Fim do tempo da sondagem de preâmbulos. Nenhum preâmbulo detectado
3	Preâmbulo detectado
4	Preâmbulo recebido
5	Nó não é membro do FCS
6	Nó é membro do FCS
7	Preâmbulo enviado. Destino final
8	Preâmbulo enviado. Nó retransmissor
9	Confirmação ao preâmbulo não foi detectada. Canal inativo
10	Confirmação ao preâmbulo detectada
11	Confirmação ao preâmbulo recebida
12	Momento de receber os dados alcançado
13	Nó atual é o destino final
14	Nó atual não é o destino final
15	Tempo limite para a colisão iminente
16	Momento de receber os dados alcançado
17	Mensagem de dados recebida
18	Mensagem de dados pendente para transmissão
19	Canal livre
20	Preâmbulo enviado
21	Confirmação ao preâmbulo não detectada
22	Confirmação ao preâmbulo detectada
23	Confirmação ao preâmbulo recebida
24	Fim do tempo τ
25	Mensagem de dados enviada

pertencente ao FCS. Quando um membro do FCS entra em atividade e recebe o preâmbulo, ele começa a enviar seus próprios preâmbulos à frente. O primeiro preâmbulo funciona como uma mensagem eACK, ou seja, uma confirmação ao nó anterior que o preâmbulo foi recebido. Se não houver nenhum membro do FCS em atividade no intervalo destinado à resposta do preâmbulo, a transição 21 retorna o protocolo para o estado EP. Se um sinal de resposta é detectado no canal, o protocolo muda para o estado RC (recebendo a mensagem de confirmação), através da transição 22. Depois disso, a transição 23 leva o protocolo para o estado AG (aguardando τ). Neste estado, espera-se o tempo definido τ para enviar a mensagem de dados. Depois disso, a transição 24 leva ao estado ED (enviando a mensagem de dados). No final da transmissão, o protocolo retorna ao estado IN pela transição 25.

Quando não há mensagens para serem enviadas, o protocolo alterna entre os estados IN e SP. Entretanto, quando um preâmbulo é detectado no estado SP, o protocolo altera o estado para RP (recebendo preâmbulo) através da transição 3. Depois de receber o preâmbulo de um

nó vizinho, chega-se ao estado VF (verificando o FCS) pela transição 4. Neste estado, o nó que detectou o preâmbulo verifica se pertence ao FCS do nó transmissor. Se for esse o caso, o estado muda para $EP2$ pela transição 6, caso contrário o nó retorna para IN pela transição 5. Se o nó é o destino final da mensagem, envia de volta um preâmbulo com um FCS vazio, apenas para sinalizar que recebeu o preâmbulo. Depois disso, o nó vai para o estado IE pela transição 7 e aguarda a mensagem de dados. Se o nó não é o destino final da mensagem, assume a função de retransmissor e começa a enviar preâmbulos para encontrar o próximo transmissor na rota. O primeiro preâmbulo enviado atua ao mesmo tempo como um eACK para o nó do salto anterior. Depois de enviar um preâmbulo, a transição 8 leva ao estado $SC2$ que espera por uma resposta de um dos seus membros do FCS. Os estados EP e $EP2$ são semelhantes em suas atribuições, a diferença é que no primeiro, o preâmbulo está sendo enviado pelo nó de origem, enquanto que no segundo, o preâmbulo está sendo enviado durante a retransmissão entre nós intermediários.

Quando o nó está aguardando uma resposta no estado $SC2$, e nenhum sinal é detectado no canal, o protocolo retornará para $EP2$, pela transição 9. Caso uma resposta é detectada, a transição 10 direciona o protocolo para o estado $RC2$ no qual o primeiro preâmbulo do próximo nó está sendo recebido. Depois desta recepção, o protocolo é direcionado para o estado IE pela transição 11. Neste estado, o rádio é desativado até o instante da recepção da mensagem de dados. Alcançado este momento, o protocolo atinge o estado RD pela transição 12. Neste estado, a mensagem de dados é recebida pelo nó. Após esta recepção, o endereço de destino final é verificado. Se a mensagem é endereçada ao nó receptor, a mensagem é enviada para a camada superior e o protocolo retorna ao estado inativo IN pela transição 13. Se o nó receptor não for o destino, a mensagem é transmitida para o próximo nó retransmissor, alcançando o estado ED pela transição 14.

Quando uma colisão iminente é identificada no estado $SC2$, a transição 15 direciona a máquina de estados para o estado $IE2$ para evitar a colisão entre dados e preâmbulo no salto anterior. Neste estado, o nó interrompe a transmissão de preâmbulos e aguarda a chegada da mensagem de dados. A transição 16 leva o protocolo para o estado $RD2$ no qual a mensagem de dados é recebida. Depois disso, o nó retransmissor assume o papel do primeiro nó transmissor (nó de origem) e vai para o estado CS pela transição 17 e o processo de envio de dados é reiniciado. Como o envio dos preâmbulos foi interrompido, a transição 17 é responsável por reiniciar estas transmissões assim como é feito no nó de origem.

4 APA-MAC

Este capítulo apresenta o protocolo *cross-layer* assíncrono *anycast* proposto neste trabalho, intitulado APA-MAC (*Advanced Preamble Anycast for MAC layer*). O protocolo é baseado no PAX-MAC (HEIMFARTH et al., 2016), entretanto, utiliza técnicas que autoriza o envio da mensagem de dados somente se o nó receptor identificar que o canal está livre. Este procedimento previne a colisão iminente sem interromper o envio dos preâmbulos, evitando o aumento da latência causada pelo reinício do processo de transmissão da mensagem de dados. APA-MAC também emprega técnicas para acelerar o avanço dos preâmbulos enquanto otimiza o número de saltos curtos. Estas características concedem ao APA-MAC menor latência, mesmo em cenários com transmissões concorrentes.

4.1 Visão Geral

APA-MAC é um protocolo MAC *cross-layer* que integra funções da camada de rede e da camada MAC. Visando reduzir a latência em protocolos MAC assíncronos, utiliza comunicação *anycast* e o roteamento geográfico.

Uma vez que os nós operam em ciclos de trabalho (hora está ativo, hora inativo) e não possuem sincronização com outros nós na rede, quando um nó tem uma mensagem para transmitir, envia antes uma série de preâmbulos curtos para encontrar o nó do próximo salto. A mensagem é retransmitida pelos nós intermediários até chegar ao nó no qual a mensagem é destinada. Para reduzir o tempo no estabelecimento da comunicação entre os nós, APA-MAC emprega a comunicação *anycast*. Neste tipo de comunicação, é definido um grupo de nós vizinhos habilitados a receber a mensagem. O primeiro nó deste grupo que acordar, envia uma mensagem de confirmação (eACK) para o nó transmissor interrompendo a sequência de preâmbulos.

Assim como o PAX-MAC (HEIMFARTH et al., 2016), no protocolo APA-MAC, diferente de outros protocolos *anycast*, a mensagem de dados não é enviada logo após receber uma mensagem eACK. O preâmbulo propaga avançando pelos nós intermediários, criando uma reserva de tempo para a recepção dos dados. Dessa forma, ao mesmo tempo em que os preâmbulos estão saltos adiante programando os próximos nós transmissores, os dados estão sendo transmitidos.

Caso ocorra a aproximação da mensagem de dados à transmissão de preâmbulos (colisão iminente), o protocolo PAX-MAC interrompe o envio dos preâmbulos e espera até a chegada

da mensagem de dados, evitando uma colisão entre dados e preâmbulo. Portanto, um atraso inicial é mantido pelo nó de origem, para reduzir a probabilidade de que este evento ocorra. Entretanto, no protocolo APA-MAC, a mensagem de dados é enviada somente após o nó do próximo salto realizar um *carrier sensing*, e enviar uma mensagem autorizando o envio caso o canal esteja desocupado. Assim, o envio de preâmbulos não precisa ser interrompido, uma vez que não há risco de uma colisão iminente com os dados. Dessa forma não há necessidade de manter um atraso inicial para o encaminhamento dos dados.

No protocolo PAX-MAC, o tempo para a recepção dos dados informado no preâmbulo, é incrementado a cada salto, pelo tempo de transmissão da mensagem de dados. Assim, cada nó reprogramado, entra em atividade no momento da recepção dos dados. APA-MAC também utiliza o mesmo princípio, entretanto quando os nós reprogramados entram em atividade, antes da recepção dos dados, enviam a mensagem autorizando o envio. Portanto, a cada salto, o tempo para a recepção dos dados informado pelo preâmbulo, é incrementado pelo tempo da transmissão da autorização mais o tempo de transmissão dos dados.

Neste trabalho, a quantidade de tempo que é incrementada nos preâmbulos em cada salto, define o conceito de períodos. Portanto, o tempo informado pelo preâmbulo, indica o início do próximo período, e quando este tempo é incrementado, define-se então o início do período posterior. Dessa forma, se o nó do salto S_1 está programado para receber os dados no início do período 1, o nó de salto $S + n$ está programado para recebê-lo no início do período n .

Os preâmbulos carregam a informação da quantidade de tempo para o início do próximo período. Essa informação é atualizada em cada preâmbulo enviado. Dessa forma, quando um membro do FCS receber um preâmbulo, receberá a leitura mais recente da porção do tempo. O início desse período, também é o instante em que o nó do salto anterior, estará aguardando a autorização para enviar a mensagem de dados. Para realizar o *carrier sensing* antes do envio da autorização, o nó que receberá os dados, sempre entrará em atividade instante antes do início do período.

O funcionamento do APA-MAC, é demonstrado na Figura 4.1 que apresenta um exemplo de transmissão de mensagens. Neste exemplo, a origem S inicia uma transmissão endereçada ao destino D . A figura representa a transmissão nos quatro primeiros saltos. Para cada salto, o FCS é apresentado. Antes do envio da mensagem de dados, ocorre a transmissão de uma série de preâmbulos no nó de origem, intercalados com períodos de escuta para recepção do eACK.

Os preâmbulos informam para o próximo transmissor, quando começará o período $P2$. Como o nó 1, membro do FCS, acorda primeiro, ele responde com uma mensagem eACK e assume o papel de próximo transmissor. Sabendo agora o tempo em que começará o período $P2$, imediatamente, o nó 1 inicia a transmissão da série de preâmbulos endereçadas ao seu FCS, informando quando será o início do período $P3$.

Antes de enviar o segundo preâmbulo, o nó 1 avalia que não há mais tempo de enviar o preâmbulo, aguardar um eACK e ainda realizar a sondagem do canal antes do período $P2$. Antes de enviar a mensagem autorizando o envio dos dados, o nó 1 deve realizar um *carrier sensing* e verificar se o canal está desocupado. Como a sondagem pode começar logo no início da espera por um eACK numa sequência de preâmbulos, sua duração deve ser o tempo de espera por um eACK mais o tempo de transmissão de um preâmbulo. Uma vez que o nó 1 não receberá mais a mensagem de dados em $P2$, reprogramará a tentativa para $P3$. Sendo assim, a mensagem de dados estará disponível para o nó do próximo salto somente em $P4$. Portanto, a partir do segundo preâmbulo, será informado ao FCS (formado pelos nós 3 e 4), que o período do próximo transmissor começará em $P4$ e não mais em $P3$. O nó 3, é o primeiro a acordar e receber o preâmbulo informando o período $P4$. Logo em seguida, começa sua sequência de preâmbulos endereçadas ao seu FCS (formado pelos nós 5 e 6), definindo o período posterior $P5$.

Aproximando o período $P2$, o nó 1 acorda para realizar a sondagem do canal. Neste momento, o nó 3 está recebendo um eACK de 5, membro eleito do FCS. A transmissão do nó 5, não alcança 1, que por sua vez, poderá enviar a mensagem autorizando 5 transmitir os dados. A partir desse momento, ocorrem transmissões simultâneas de dados e preâmbulos.

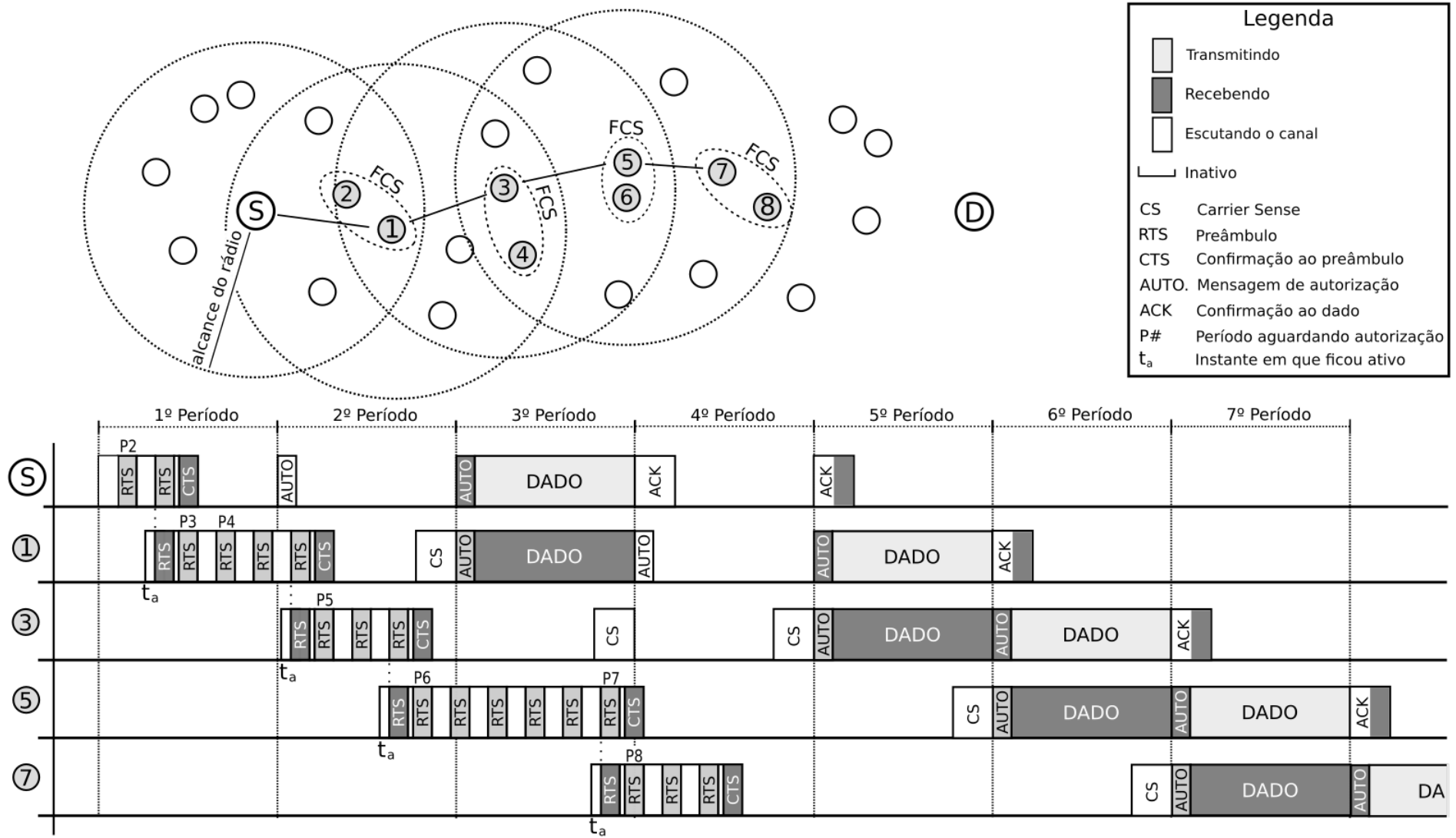
Conforme programado, aproximando o período $P4$, o nó 3 irá realizar a sondagem do canal. Entretanto, o nó 5 ainda não encontrou o próximo transmissor e está transmitindo preâmbulos. Portanto, o canal está ocupado e não será enviado a mensagem de autorização. Sendo assim, o nó 3 irá realizar uma nova tentativa no próximo período. O nó 1 não recebeu autorização e portanto, também irá aguardar novamente no próximo período. Sabendo o início de $P5$, o nó 5 consegue calcular o período anterior $P4$. Sendo assim, está ciente que seus preâmbulos interferiram no momento em que o nó do salto anterior estaria sondando o canal. Consequentemente, irá acontecer o atraso de um período para que o nó do salto anterior receba a mensagem de dados. Portanto, o nó 5 irá passar a informar em seus preâmbulos, que o início do período do próximo transmissor ocorrerá em $P7$, ao invés de $P6$.

Nó início do período $P4$, após o envio da mensagem de dados, S ficará aguardando a confirmação que os dados foram recebidos. Se o nó 1 recebeu os dados, ele será enviado para o nó do salto seguinte. Desta forma, a própria transmissão dos dados, servirá como mensagem de confirmação para S . Entretanto, o canal não estava livre para o nó 3, portanto, o nó 1 não obteve a permissão para enviar a mensagem de dados. Consequentemente, S não recebeu a confirmação e irá aguardar novamente, no início do próximo período. Caso houvesse algum tipo de colisão e o nó 1 não recebesse os dados, no próximo período 1 enviaria novamente a S uma mensagem autorizando a transmissão dos dados. Desta forma, S perceberia que houve problema na transmissão da mensagem de dados e a enviaria novamente.

No início do período $P5$, o canal está livre para o nó 3, que envia para o nó 1, a mensagem autorizando a transmissão dos dados. Consequentemente, 1 enviará os dados para 3, e essa transmissão também servirá para S , como uma confirmação que a mensagem de dados foi recebida por 1.

A partir do período $P5$, ocorrem as transmissões dos dados através dos nós 1, 3, 5 e 7. Cada transmissão é constituída pelas etapas de espera pela autorização, envio dos dados e espera pela confirmação de recebimento. O processo de envio de preâmbulos e dados, continua até que alcancem o destino D .

Figura 4.1 – Visão geral do protocolo APA-MAC



4.2 Critério de Seleção do FCS

O critério de seleção dos nós candidatos ao encaminhamento de uma mensagem (FCS), baseia-se nas regras do protocolo PAX-MAC, expostas na Seção 3.1.1. A seleção dos nós que promovem o maior progresso até o destino, reduz o número total de saltos, e conseqüentemente, diminui a latência total. Para selecionar os membros do FCS, os nós sensores possuem conhecimento de suas próprias posições geográficas, além das posições geográficas dos nós vizinhos e do destino.

A decisão de quais nós que irão pertencer ao FCS é tomada no lado do nó emissor. Esta informação está contida nos preâmbulos que serão enviados. Somente os membros do FCS são habilitados a enviar um eACK em resposta a um preâmbulo.

Ao receber um preâmbulo informando o destino final da mensagem, os nós calculam a distância de seus vizinhos até o destino e os ordena com base na distância. Desta forma, se o nó quer escolher n nós para seu FCS, informa no preâmbulo a distância do n -ésimo nó mais próximo do destino. Quando um nó recebe o preâmbulo, ele calcula sua distância até o destino final. Se o valor for menor ou igual a distância informada, o nó se elege como membro do FCS. Essa técnica mantém o tamanho do preâmbulo reduzido, pois independente de quantos membros possui o FCS, o tamanho do preâmbulo será o mesmo.

Após receber um preâmbulo, o nó membro do FCS, aguarda um tempo antes de enviar seu primeiro preâmbulo. Durante este tempo, o nó permanece sondando o canal verificando se algum membro do FCS já respondeu ao preâmbulo. Se identificar uma resposta, significa que outro nó já será escolhido, e portanto, não precisará responder ao preâmbulo. Nós que promovem maior progresso, possuem maior prioridade. Desta forma, se mais de um membro do FCS receber o preâmbulo, o mais perto do destino deverá responder primeiro. O tempo de espera até enviar o preâmbulo, é demonstrado na equação 4.1.

$$t = \left(1 - \frac{d_e - d_r}{r}\right) \times \frac{t_a}{2} \quad (4.1)$$

Em que t é o tempo de espera para enviar o primeiro preâmbulo, d_e a distância do emissor até o destino, d_r a distância do receptor até o destino, r o alcance do rádio e t_a o tempo de espera pelo eACK. Desta forma, conforme a distância do nó membro do FCS até o destino, o primeiro preâmbulo será enviado, no pior caso, na metade do tempo de espera por um eACK.

4.3 Configuração da Cardinalidade do FCS

Através da Equação 3.1 da Seção 3.1.1, é possível encontrar a cardinalidade padrão do FCS para que o tempo médio de encontrar o nó do próximo salto, seja igual ao tempo de transmissão dos dados. PAX-MAC adota esta cardinalidade em seu FCS, de forma que preâmbulos e dados podem avançar ao mesmo tempo, na mesma velocidade, em locais diferentes da rede. Entretanto, a velocidade de avanço dos preâmbulos não é constante, possibilitando que a mensagem de dados se aproxime dos preâmbulos. Neste caso, para realizar a transmissão dos dados, é necessário esperar por um ou mais períodos até que o canal esteja livre e o envio dos dados seja autorizado, o que aumenta a latência total.

Com o objetivo de que a transmissão dos preâmbulos não atrase o avanço dos dados, foram propostos dois métodos para acelerar o avanço dos preâmbulos. No primeiro, a cardinalidade padrão é multiplicada e este valor será fixo, sendo usado em toda transmissão dos dados. No segundo, a cardinalidade padrão também é multiplicada, mas o multiplicador estabelecido decresce conforme os preâmbulos se distanciam dos dados. O primeiro método é nomeado como FCS fixo, e o segundo como FCS variável.

As Figura 4.2 e 4.3 exemplificam o FCS fixo e o FCS variável respectivamente. Conforme as figuras, o nó S aguarda a mensagem de autorização para o envio dos dados, e para cada salto é demonstrado qual seria o tamanho do FCS caso o respectivo salto estivesse com a transmissão de preâmbulos. A cardinalidade padrão em ambos exemplos foi 4. Para o exemplo do FCS fixo, foi utilizado o multiplicador $1,5x$, dessa forma como ilustrado, o FCS terá sempre 6 elementos. Para o FCS variável, foi utilizado o multiplicador $2x$ que decresce num intervalo de 4 períodos até atingir um valor multiplicativo mínimo de $0,5x$. Dessa forma, a cardinalidade do FCS irá variar de 8 a 2 elementos, quando a mensagem de dados estiver entre 1 a 4 períodos de distância. Se os preâmbulos distanciarem mais que 4 períodos, será mantido o multiplicador mínimo. Dessa forma, quando ocorrer a aproximação dos dados, a velocidade de avanço dos preâmbulos será aumentada gradativamente, enquanto que ao distanciar, a velocidade será reduzida, mas no entanto, provendo saltos maiores.

Uma vez que o número de elementos do FCS variável está relacionado com o tempo para a recepção dos dados, o cálculo da cardinalidade é realizado antes de cada preâmbulo enviado. Este cálculo é demonstrado na Equação 4.2.

Figura 4.2 – FCS Fixo

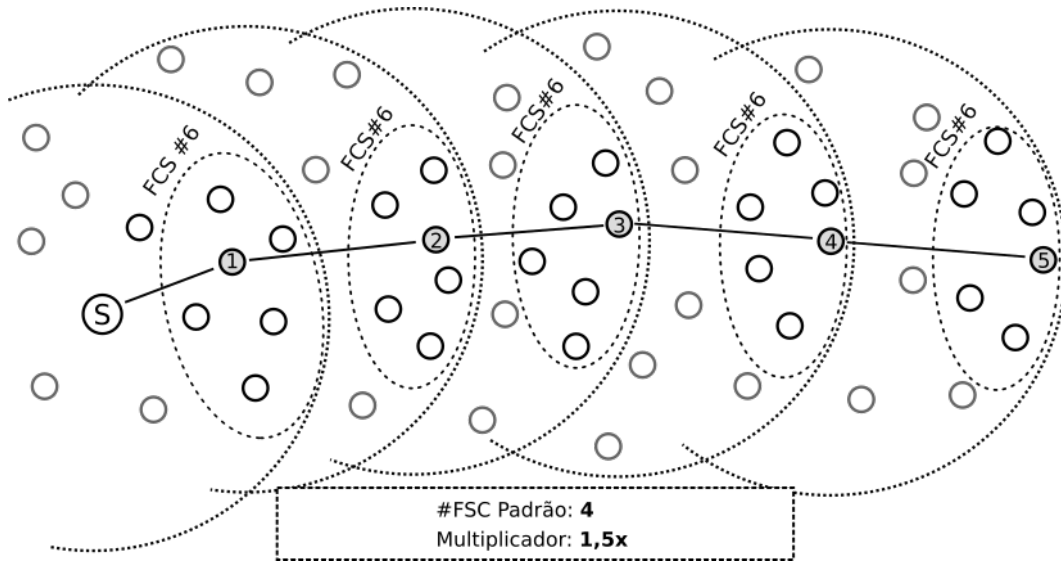
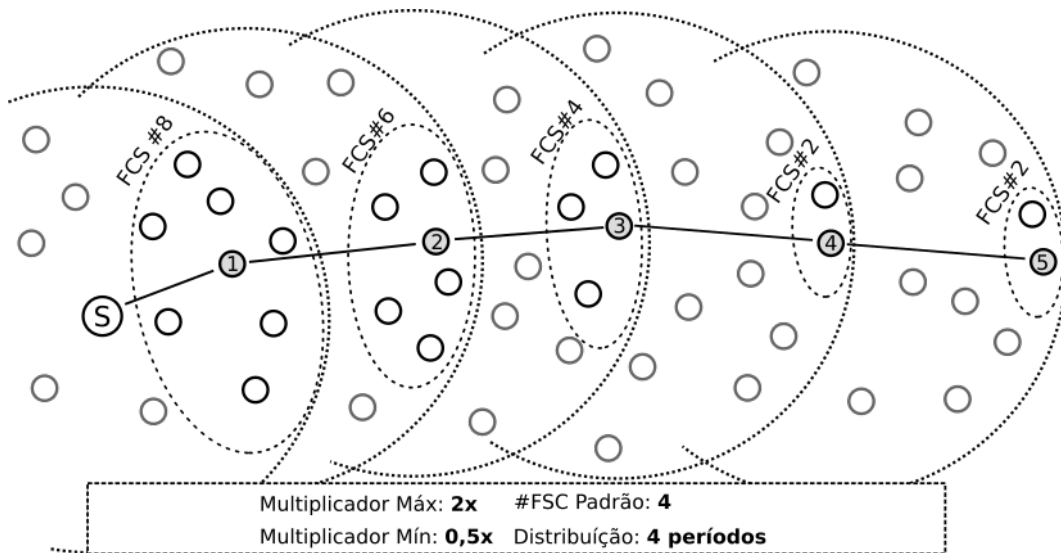


Figura 4.3 – FCS Fixo



$$v = v_p \times \begin{cases} m, & \text{para } 0 \leq t \leq t_{\text{periodo}} \\ \left(\frac{m-l_m}{l_t-1}\right) \times (l_t - t) + l_m & \text{para } t_{\text{periodo}} < t < l_t \\ l_m, & \text{para } t \geq l_t \end{cases} \quad (4.2)$$

Em que v é a cardinalidade que deverá ser usada, v_p a cardinalidade padrão, m o multiplicador, l_m o limite mínimo do multiplicador, t o tempo restante para a recepção dos dados, t_{periodo} o tempo de um período e l_t o tempo limite da variação do FCS, ou seja, tempo em que o FCS se torna fixo com o multiplicador mínimo l_m . Dessa forma, aplicando a equação ao exemplo da Figura 4.3, teria-se $v_p = 4$, $m = 2$, $l_m = 0,5$ e $l_t = 4 \times t_{\text{periodo}}$.

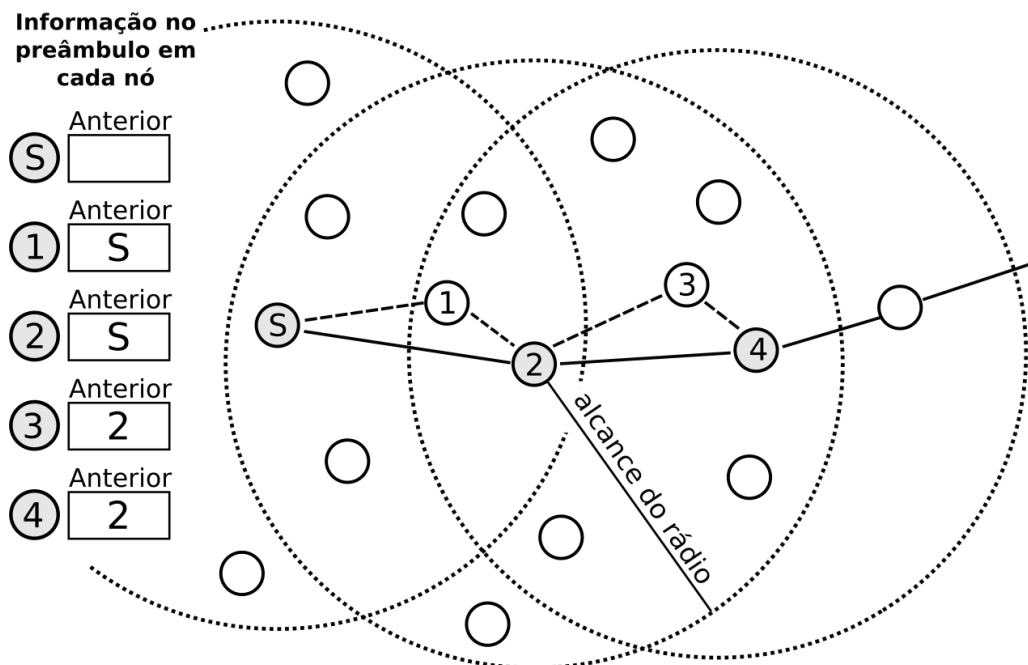
4.4 Otimização de saltos

Os preâmbulos não se propagam a uma velocidade constante, portanto em alguns casos a mensagem de dados pode alcançar a área de transmissão dos preâmbulos. Neste caso, o nó que possui os dados não receberá autorização para enviá-lo, uma vez que o nó do salto seguinte não encontra o canal desocupado. Uma nova tentativa de transmissão dos dados acontecerá somente no início do próximo período, aumentando a latência total.

Para o protocolo atingir sua melhor performance na latência, a mensagem de dados deve ser sempre enviada no período seguinte após o seu recebimento. Portanto, para diminuir o tempo de contato com o próximo transmissor, é necessário aumentar a cardinalidade do FCS. Esta ação acelera a propagação dos preâmbulos, tornando-se menos provável a aproximação dos dados. Entretanto, aumentando o FCS, nem sempre os nós que promovem o maior avanço serão eleitos. Isto pode ocasionar mais saltos necessários para a mensagem chegar até o destino final, também aumentando a latência total.

Com o objetivo de reduzir o efeito da latência causada pelo aumento no número de saltos, APA-MAC utiliza uma técnica exemplificada na Figura 4.4.

Figura 4.4 – Otimização de saltos



No preâmbulo é informado qual é o nó pertencente ao salto anterior ao nó atual. Portanto, quando um nó recebe o preâmbulo, é capaz de identificar se o nó do salto anterior ao emissor, também é seu vizinho. Conforme exemplificado na figura, o nó *S* começa o envio da

sequência de preâmbulos, e 1, membro do FCS, é o primeiro nó a receber o preâmbulo. Como não consta informação do transmissor anterior, se trata do nó de origem e portanto não existe saltos anteriores. Imediatamente, 1 começa o envio da sequência de preâmbulos, informando que o nó do salto anterior é S . O nó 2, é o primeiro membro do FCS de 1 a receber o preâmbulo. Ele é ciente que S é um nó vizinho, sendo assim 1 não é mais necessário. Portanto, na sequência de preâmbulos transmitidas por 2, será informado que o nó do salto anterior é S , e não 1. Quando 1 receber o primeiro preâmbulo, durante a espera pelo eACK, perceberá que não foi considerado como o nó do salto anterior, e portanto, não fará parte da transmissão dos dados. O mesmo processo ocorre nos saltos seguintes. O nó 3, membro do FCS de 2, recebe o preâmbulo, e define o nó 2 como nó do salto anterior, já que não alcança S . Entretanto, durante a transmissão da sequência de preâmbulos pelo nó 3, o nó 4, eleito do FCS, é ciente que alcança 2. Logo, irá defini-lo como o nó do salto anterior. Quando 3 receber o primeiro preâmbulo enviado por 4, será notificado que não foi escolhido como o nó do salto anterior e não participará da transmissão dos dados.

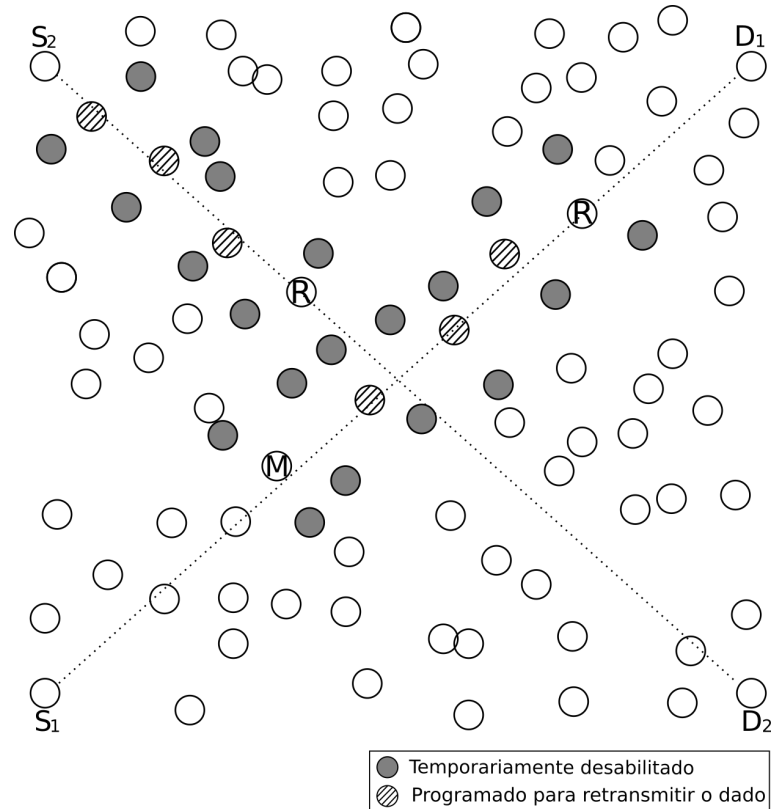
4.4.1 Proteção do canal

Com a finalidade de manter a performance na taxa de entrega e na latência total em cenários com alto tráfego de mensagens, APA-MAC adota mecanismos para evitar colisões entre diferentes fluxos de transmissão. A Figura 4.5 apresenta o esquema de proteção do canal de comunicação.

Os nós $S1$ e $S2$ são os nós de origem de duas transmissões simultâneas diferentes, e os nós $D1$ e $D2$ os respectivos destinos. Eventualmente, uma mensagem de um fluxo pode colidir com a mensagem concorrente. Trata-se de uma colisão entre fluxos diferentes. Os nós rotulados com R estão sondando o caminho da origem para o destino através da emissão de preâmbulos. Os nós identificados com M estão no momento com a posse dos dados. Os nós hachurados foram selecionados para formar o caminho para o destino e são programados para receber e retransmitir a mensagem de dados.

Quando algum nó recebe um preâmbulo e verifica que não faz parte do FCS, ele permanece temporariamente desativado até a passagem da mensagem de dados, calculada conforme as informações do preâmbulo. Os nós temporariamente desabilitados são programados para evitar a execução do ciclo de trabalho até que a mensagem de dados tenha sido transmitida com sucesso pelos nós retransmissores próximos. Os nós desabilitados atuam como uma bar-

Figura 4.5 – Mecanismo de Proteção do Canal



reira, não respondendo os preâmbulos de outros fluxos de comunicação, impedindo possíveis colisões. Os nós desativados são representados na figura preenchidos em cinza.

No exemplo representado pela Figura 4.5, o nó R no caminho $S_2 \rightarrow D_2$, não receberá uma resposta ao preâmbulo dos membros do FCS, uma vez que eles estão temporariamente desativados. Esta condição é mantida até que a mensagem de dados M tenha atravessado pelos nós nas proximidades.

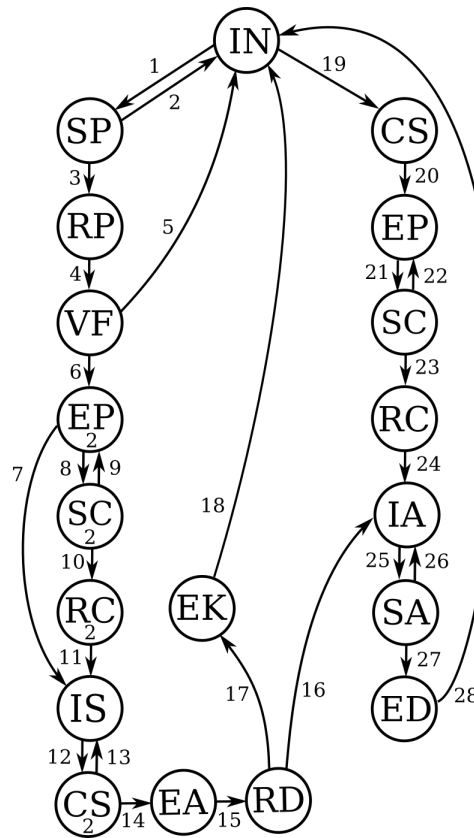
4.4.2 Máquina de estados

Esta seção descreve a máquina de estados do protocolo APA-MAC ilustrado na Figura 4.6. Os Quadros 4.1 e 4.2 apresentam os significados dos estados e suas transições respectivamente.

Durante os ciclos de atividade, cada nó alterna entre os estados IN (inativo) e SP (sondando o canal em busca de preâmbulos) através das transições 1 e 2. O ciclo de trabalho determina o tempo de SP em relação ao tempo total do ciclo.

Ao receber a mensagem da camada superior, inicia-se uma nova transmissão, direcionando ao estado CS (*carrier sensing*) pela transição 19. Se o canal estiver livre, a transição 20

Figura 4.6 – Máquina de estados do protocolo APA-MAC



Quadro 4.1 – Lista dos estados da máquina de estados

Estado	Descrição
IN - Inativo	Nó está inoperante
SP - Sondando preâmbulos	Escutando o canal em busca de preâmbulos
CS - <i>Carrier sensing</i>	Escutando o canal em busca de atividade
EP - Enviando preâmbulo	Preâmbulo está sendo enviado
SC - Sondando confirmação	Escutando o canal em busca da mensagem de confirmação
RC - Recebendo confirmação	A mensagem de confirmação está sendo recebida
IA - Inativo aguardando autorização	Nó fica inativo até a hora de receber a autorização
SA - Sondando autorização	Escutando o canal em busca da mensagem de autorização
ED - Enviando dados	Enviando a mensagem de dados
RP - Recebendo preâmbulo	Preâmbulo está sendo recebido
VF - Verificando FCS	Verificando se o nó é um membro do FCS
IS - Inativo aguardando sondagem	Nó fica inativo até a hora de realizar a sondagem do canal
EA - Enviando autorização	Enviando a mensagem de autorização
RD - Recebendo dados	A mensagem de dados está sendo recebida
EK - Enviando confirmação dos dados	Enviando a mensagem confirmado que os dados foram recebidos

direcionará para o estado EP (enviando preâmbulo). Neste estado, o preâmbulo é transmitido em *broadcast*. Pela transição 21, o estado é alterado para SC (sondando a mensagem de confirmação). Neste estado, o nó aguarda uma resposta eACK de um vizinho membro do FCS.

Quadro 4.2 – Lista das transições dos estados da máquina de estados

Transição	Descrição
1	Fim do tempo no modo inativo
2	Fim do tempo da sondagem de preâmbulos. Nenhum preâmbulo detectado
3	Preâmbulo detectado
4	Preâmbulo recebido
5	Nó não é membro do FCS
6	Nó é membro do FCS
7	Preâmbulo enviado. Destino final
8	Preâmbulo enviado. Nó retransmissor
9	Confirmação ao preâmbulo não foi detectada. Canal inativo
10	Confirmação ao preâmbulo detectada
11	Confirmação ao preâmbulo recebida
12	Momento de sondar atividade no canal alcançado
13	O canal está ocupado
14	O canal está desocupado
15	Mensagem de autorização enviada
16	Mensagem de dados recebida. Destino final
17	Mensagem de dados recebida. Nó retransmissor
18	Mensagem de confirmação aos dados enviados
19	Mensagem de dados pendente para transmissão
20	O canal está desocupado
21	Preâmbulo enviado
22	Confirmação ao preâmbulo não detectada
23	Confirmação ao preâmbulo detectada
24	Confirmação ao preâmbulo recebida
25	Tempo para receber a mensagem de autorização alcançado
26	Mensagem de autorização não detectada
27	Mensagem de autorização detectada
28	Mensagem de dados enviada

Quando o primeiro membro do FCS acorda e recebe o preâmbulo, começa a enviar seus próprios preâmbulos em busca do próximo transmissor. O primeiro preâmbulo funciona como uma mensagem, eACK, ou seja, uma confirmação ao nó anterior que o preâmbulo foi recebido. Se nenhum membro do FCS estiver acordado no momento do envio do preâmbulo, sem um eACK, a transição 22 retorna o protocolo para o estado EP. Entretanto, se um sinal de resposta é detectado no canal, o protocolo muda para o estado RC (recebendo a mensagem de confirmação), através da transição 23. Em sequência, a transição 24 leva o protocolo para o estado IA (inativo aguardando autorização).

Os preâmbulos carregam a informação do tempo em que o nó emissor estará aguardando a mensagem autorizando o envio dos dados. Portanto, após receber a confirmação que o preâmbulo foi recebido, o nó ficará inativo até o momento informado. Dessa forma, no instante da

transmissão, a transição 25 irá direcionar o protocolo para o estado SA (sondando mensagem de autorização) para a recepção da mensagem de autorização. Se o canal de comunicação não está livre, o nó do próximo salto não enviará a autorização, e a transição 26, direcionará o protocolo novamente para o estado IA. Neste caso, no próximo período, será realizada outra tentativa para receber a mensagem de autorização. O intervalo entre os períodos é equivalente ao tempo da transmissão da mensagem de autorização mais a mensagem de dados. Mas, se durante o estado SA, o nó receber a mensagem de autorização, a transição 27, encaminhará o nó para o estado ED (enviando mensagem de dados). Após a transmissão dos dados, o protocolo retorna ao estado IN pela transição 28.

Quando não há mensagens para serem enviadas, o protocolo alterna entre os estados IN e SP. Entretanto, quando um preâmbulo é detectado no estado SP, o protocolo altera o estado para RP (recebendo preâmbulo) através da transição 3. Depois de receber o preâmbulo de um nó vizinho, chega-se ao estado VF (verificando o FCS) pela transição 4. Neste estado, o nó que recebeu o preâmbulo verifica se pertence ao FCS do nó transmissor. Caso pertença ao FCS, o estado muda para EP2 pela transição 6, caso contrário o nó retorna para IN pela transição 5. Se o nó é o destino final da mensagem, envia de volta um preâmbulo com um FCS vazio, apenas para sinalizar que recebeu o preâmbulo. Depois disso, o nó vai para o estado IS (inativo aguardando o momento de sondar o canal) pela transição 7, e aguarda o momento de realizar o *carrier sensing* para que possa enviar a mensagem autorizando o envio dos dados. Se o nó não é o destino final da mensagem, assume a função de retransmissor e começa a enviar preâmbulos para encontrar o próximo nó na rota. O primeiro preâmbulo enviado atua ao mesmo tempo como um eACK para o nó do salto anterior. Depois de enviar um preâmbulo, a transição 8 leva ao estado SC2 que espera por uma resposta de um dos seus membros do FCS.

Os estados EP e EP2 são semelhantes em suas atribuições, a diferença é que no primeiro, o preâmbulo está sendo enviado pelo nó de origem, enquanto que no segundo, o preâmbulo está sendo enviado durante a retransmissão entre nós intermediários. Em ambos os estados, momento antes do envio de cada preâmbulo, é verificado se para o período informado, ainda há tempo para enviar o preâmbulo e receber a confirmação, além do tempo para que o nó do próximo salto realize a sondagem do canal, autorizando o envio dos dados. Se o tempo restante é insuficiente, o período em que o nó estará aguardando a mensagem de confirmação, será incrementado, e as informações do preâmbulo, atualizadas.

Durante o estado *SC2*, se o nó está aguardando uma resposta *eACK*, e nenhuma mensagem é detectada no canal, o protocolo retorna para *EP2*, pela transição 9. Se uma resposta é detectada, a transição 10 direciona o protocolo para o estado *RC2* no qual o primeiro preâmbulo do próximo nó transmissor está sendo recebido. Depois da recepção, o protocolo é direcionado para o estado *IS* pela transição 11. Neste estado, o rádio é desativado até o instante de realizar o *carrier sensing* no canal. Alcançado este momento, o protocolo atinge o estado *CS2* (*carrier sensing*) pela transição 12. Se o canal está desocupado, a transição 14 direciona o protocolo para o estado *EA* (enviando autorização), para que a mensagem autorizando o envio dos dados seja enviada. Entretanto, se o canal está ocupado, o protocolo retornará para o estado *IS*, pela transição 13. Neste caso, será realizada outra tentativa, momento antes do próximo período. Uma vez que o nó do salto anterior não receberá a mensagem de autorização, também realizará outra tentativa no próximo período, assim como já mencionado na transição 26.

Quando a mensagem de autorização é enviada para o nó do próximo salto, a transição 15 direciona o protocolo para o estado *RD* (recebendo os dados). Se o nó que recebeu os dados não é o destino final, a transição 16 irá direcionar o protocolo para o estado *IA*, onde permanecerá inativo até momento de receber a mensagem autorizando o envio dos dados. As próximas transições ocorrerão da mesma forma como já descrito anteriormente. Entretanto, se o nó é o destino final da mensagem, a transição 17 irá direcionar para o estado *EK* (enviando confirmação dos dados). A mensagem confirmando o recebimento dos dados, é enviada somente pelo nó do destino final, uma vez que o nó do salto anterior não ouvirá a retransmissão dos dados. Após enviar a confirmação, a mensagem de dados é enviado para a camada superior e o protocolo retorna ao estado inativo *IN* pela transição 18.

5 Resultados e Discussões

Esta seção apresenta as configurações usadas na simulação e os resultados e discussões da avaliação do protocolo proposto neste trabalho. Foram avaliadas as métricas como latência, consumo e taxa de entrega, também considerando cenários propícios à colisões.

5.1 Configurações da Simulação

O protocolo APA-MAC, assim como todos protocolos testados, foram desenvolvidos no simulador *GrubiX Wireless Network Simulator*, uma reformulação e extensão do simulador ShoX (LESSMANN; HEIMFARTH; JANACIK, 2008). O simulador permite a execução de testes com resultados precisos, com cenários e configurações de rede controláveis, para uma comparação justa entre os protocolos analisados.

Para as simulações, foi adotado um modelo de rádio baseado no padrão IEEE 802.15.4 (IEEE STANDARDS ASSOCIATION, 2015), potência de transmissão fixa, *links* bidirecionais e modelo de propagação em espaço livre, para difusões isotrópicas em um meio de propagação ideal. O alcance do rádio foi configurado para 40 metros.

Os nós sensores foram implantados aleatoriamente na arena de simulação e suas localizações formam um processo de Poisson. O gráfico de conexão resultante é chamado de gráfico geométrico aleatório (LI et al., 2009). Os nós eram estáticos e trocavam informações das posições no início, portanto cada nó conhecia a posição dos nós vizinhos. O protocolo MAC proposto é combinado com um protocolo de roteamento geográfico, empregando o protocolo GPRS (KARP; KUNG, 2000).

A programação do ciclo de atividade dos nós foi independente e aleatória. Para estimar o desempenho do APA-MAC, o mesmo foi comparado com os seguintes protocolos: PAX-MAC (HEIMFARTH et al., 2016), X-MAC (BUETTNER et al., 2006), X-MAC Anycast 2 (ASHRAF; VAIDYA; KRAVETS, 2011), X-MAC Anycast 6 (ASHRAF; VAIDYA; KRAVETS, 2011), GeRaF (ZORZI; RAO, 2003) e AGA-MAC (HEIMFARTH; GIACOMIN; ARAUJO, 2015). X-MAC é um protocolo MAC assíncrono bem conhecidos na literatura, enquanto que X-MAC Anycast 2 e 6 são melhorias empregando anycast com FCS de cardinalidade dois e seis como proposto em (ASHRAF; VAIDYA; KRAVETS, 2011). GeRaF é um protocolo assíncrono *anycast* padrão. AGA-MAC é um protocolo assíncrono *anycast* que leva em conta o tamanho da mensagem para reduzir a latência da rede.

Para comparar os diferentes protocolos, todos os parâmetros foram definidos com os mesmos valores. O protocolo GeRaF foi adaptado para evitar o uso de preâmbulos específicos para cada região, o que reduzia seu desempenho. Desta forma, o comportamento de GeRaF é o mesmo que do protocolo CMAC (LIU; FAN; SINHA, 2009), sem o uso de limiar (r_0). Portanto, foi chamado neste trabalho de GeRaF/CMAC.

Antes de iniciar uma transmissão, o nó de origem realiza um *carrier sensing* de $1ms$. Se não é detectada nenhuma transmissão em curso, o nó começa a enviar uma série de preâmbulos curtos, com um tempo de transmissão de $0,5ms$. Os preâmbulos são intercalados com um tempo de espera de $0,5ms$. Neste momento, o rádio estará em modo de escuta para a detecção do *eACK*. Quando um vizinho, membro do FCS, recebe o preâmbulo, ele espera no máximo $0,25ms$ para responder com uma mensagem *eACK*. Assim como descrito em 4.2, este tempo de espera varia conforme a distância do membro do FCS até o destino, de forma que quanto mais próximo do destino, menor o tempo de espera. Antes de receber a mensagem de dados, o nó realiza um *carrier sensing* por $1ms$, e se não detectar nenhuma atividade, envia a mensagem de autorização, com um tempo de transmissão de $0,4ms$.

No protocolo PAX-MAC, conforme apresentado em Heimfarth et al. (2016), o tempo inicial de espera τ é definido pela Equação 5.1. Onde $r(v)$ é o número médio de preâmbulos necessário para encontrar um membro do FCS acordado, quando a cardinalidade do FCS é v , e $(t_{pre} + t_{eACK})$, é o tempo de transmissão de um preâmbulo mais a resposta *eACK*. O multiplicador k , define quantas vezes além do tempo de avanço, o nó manterá a posse dos dados antes de enviá-lo. Segundo Heimfarth et al. (2016), o protocolo PAX-MAC obteve o melhor desempenho quando $k = 6$. Portanto, nas simulações do PAX-MAC, foi utilizado este mesmo valor de parâmetro.

$$\tau = k \cdot r(v) \cdot (t_{pre} + t_{eACK}) \quad (5.1)$$

Todos os protocolos foram avaliados em uma rede com densidade de $8 \times 10^{-3} \left(\frac{\text{nós}}{m^2}\right)$. Os parâmetros de simulação são apresentados no Quadro 5.1. Foram utilizados dois tamanhos de mensagens de dados, que possuem tempo de transmissão equivalente a 15% e 50% do tempo total do ciclo ativo/inativo. Conforme o tamanho da mensagem de dados, dada a Equação 3.1, foi calculada a cardinalidade do FCS para que o tempo de transmissão dos dados, seja igual ao tempo médio para encontrar um nó acordado no FCS. Esta cardinalidade será chamada neste trabalho de v_p . Com o objetivo de avaliar a taxa de entrega de mensagens, os teste foram

executados em cenários com e sem tráfego concorrente, isto é, múltiplos fluxos de dados. Em todos os teste, a distância entre o nó de origem e o destino final foi sempre de 650 metros.

Para obter resultados estatisticamente relevantes, cada experimento foi executado 200 vezes, considerando um nível de confiança de $\gamma = 1 - \alpha$, $\alpha = 0,05$ para o intervalo de confiança. Para cada execução, ou seja, para cada transmissão de dados iniciada pelo nó de origem, os nós foram reposicionados aleatoriamente na área de simulação. Entretanto, os nós de origem e destino, sempre mantêm uma distância entre si de 650 metros.

Quadro 5.1 – Parâmetros usados na simulação

Símbolo	Parâmetro	Valores
R	Alcance do rádio (m)	40
P	Potência do transceptor (mW)	60
SD	Distância entre origem e destino (m)	650, 1300
η	Densidade dos nós ($\frac{\text{node}}{m^2}$)	8×10^{-3}
N_p	Número máximo de preâmbulos	98
t_{cycle}	Tempo do ciclo (s)	0.1
v	Cardinalidade do FCS (em relação ao v_p)	1x até 4x
t_{data}	Duração dos dados ($\%t_{cycle}$)	15, 50
t_{pre}	Duração do preâmbulo (s)	0.512×10^{-3}
t_{eACK}	Duração do eACK (s)	0.512×10^{-3}
t_{cs}	Duração do <i>Carrier Sense</i> (s)	1.024×10^{-3}
τ	Espera para enviar os dados (PAX-MAC) (s)	$k \cdot r(v) \cdot (t_{pre} + t_{eACK})$ $k = 6$

5.2 Avaliação da Latência

Para a avaliação da técnica do FCS variável, em todos os experimentos o multiplicador decresceu até $0,6x$ (60% da cardinalidade padrão) num intervalo de 6 períodos. A cardinalidade padrão para a mensagens de dados cujo tempo de transmissão é equivalente a 15% e 50% do ciclo total (soma dos períodos ativo e inativo), é 6 e 1 respectivamente.

Os resultados da latência utilizando as técnicas de aceleração de preâmbulos, quando as mensagens de dados possuem um tempo transmissão equivalente a 15% e 50% do ciclo, são mostrados respectivamente nas Figuras 5.1 e 5.2. Os multiplicadores da cardinalidade padrão variaram de $1x$ até $4x$. Em ambas as figuras é apresentado uma linha guia indicando o desempenho do PAX-MAC, que sempre utiliza uma cardinalidade fixa igual a padrão.

A Figura 5.1, apresenta o encaminhamento de mensagens com tamanho de 15% do ciclo. Na técnica que utiliza um FCS de tamanho fixo, o multiplicador $1x$, para um FCS de tamanho

Figura 5.1 – Latência para diferentes configurações do FCS. Mensagens com tempo de transmissão igual a 15% do ciclo total

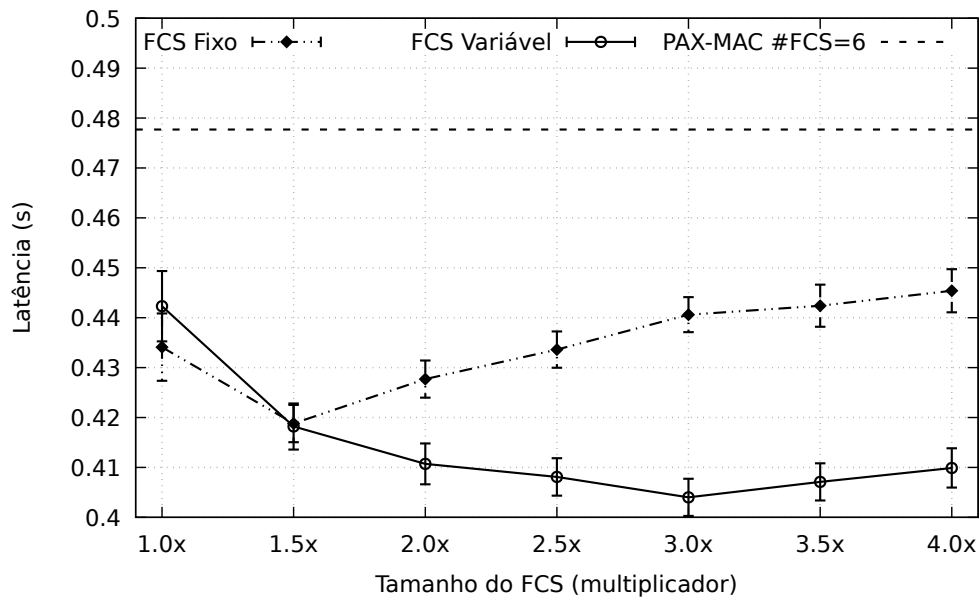
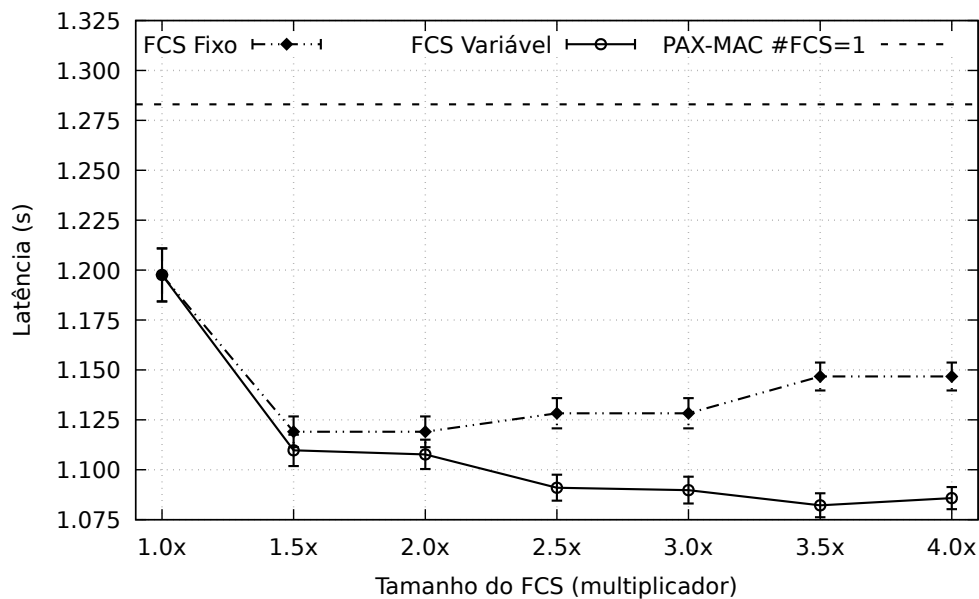


Figura 5.2 – Latência para diferentes configurações do FCS. Mensagens com tempo de transmissão igual a 50% do ciclo total



fixo, demonstra a latência obtida mantendo a cardinalidade padrão ($\#FCS = 6$), a mesma usada pelo PAX-MAC. O melhor resultado obtido foi utilizando um multiplicador igual a 1,5x, ou seja, utilizando um número de nós fixo igual a 9 (1,5 vezes a cardinalidade padrão).

A técnica de FCS variável, obteve um melhor resultado quando utilizado o multiplicador 3x. Portanto, se o tempo para a recepção dos dados for maior que 6 períodos, o FCS terá um número de nós igual 4 (0,6 vezes a cardinalidade padrão). Mas se tempo for entre 1 e 6 períodos,

a cardinalidade irá variar de 4 a 18 nós (entre 0,6 e 3 vezes a cardinalidade padrão) conforme a aproximação dos dados.

Utilizando o multiplicador $1x$, a latência obtida foi maior que na técnica que utiliza FCS fixo. Esta configuração atua de forma prejudicial no avanço dos preâmbulos, já que não promove aceleração (cardinalidade máxima será a padrão), somente desaceleração, uma vez que o número nós do FCS irá variar de 6 a 4 (entre 1 e 0,6 vezes a cardinalidade padrão).

A Figura 5.2, apresenta o encaminhamento de mensagens com tamanho de 50% do ciclo. Para a técnica que utiliza um FCS de tamanho fixo, a cardinalidade padrão é 1, portanto os multiplicadores $1,5x$, $2,5x$ e $3,5x$ irão gerar as respectivas cardinalidades 1,5, 2,5 e 3,5. Entretanto todas cardinalidades não inteiras são arredondadas para cima, portanto, os multiplicadores não inteiros foram considerados iguais ao multiplicador inteiro imediatamente superior.

O melhor resultado obtido para o FCS fixo, foi utilizando um multiplicador igual a $2x$, ou seja, utilizando um número nós fixo igual a 2 (2 vezes a cardinalidade padrão). O multiplicador $1x$, para um FCS de tamanho fixo, demonstra a latência obtida mantendo a cardinalidade padrão ($\#FCS = 1$), a mesma utilizada pelo PAX-MAC.

O melhor resultado obtido para o FCS variável, foi utilizando um multiplicador igual a $3,5x$. O multiplicador $1x$ para o FCS variável foi considerado igual ao multiplicador $1x$ para o FCS fixo, uma vez que não há valor menor que 1 para a cardinalidade.

Para ambos tamanhos de mensagens, 15% e 50% do ciclo, usando a técnica de FCS fixo com multiplicador $1x$, é possível notar que mesmo sem aumentar a cardinalidade do FCS, APA-MAC obtém menor latência em relação ao PAX-MAC. Este resultado deve-se ao fato de que APA-MAC não precisa cessar o envio de preâmbulos durante a aproximação dos dados, uma vez que a mensagem de dados é enviada somente quando recebe uma autorização de envio.

Na técnica de FCS fixo, utilizando multiplicadores acima de $1,5x$ e $2x$ respectivamente para mensagens com tamanho de 15% e 50% do ciclo, faz com que o grande número de membros no FCS promovam saltos mais curtos. Uma vez que nem sempre os nós que promovem maior progresso ao destino serão escolhidos. Nestes casos, o número maior de saltos não é compensado pela maior aceleração do preâmbulo. A otimização de saltos proposta na Seção 4.4 pode ser eficaz em até certo ponto, uma vez que atua somente nos saltos curtos, quando um nó alcança dois saltos anteriores. A otimização não atua sobre saltos que promovem avanços medianos.

Na técnica de FCS variável, a aceleração de avanço dos preâmbulos terá seu maior valor durante os primeiros saltos, quando preâmbulo e dados estão próximos. Entretanto, conforme os preâmbulos se distanciam dos dados a aceleração será reduzida, mantendo menores variações, uma vez que começará a atuar apenas mantendo a velocidade de avanço dos preâmbulos. Nesta técnica, portanto, saltos com avanços medianos irão ocorrer nos primeiros saltos. Mas, por outro lado, conforme os preâmbulos se distanciam dos dados, apenas os nós que promovem o maior progresso ao destino serão escolhidos.

Uma vez encontrado os melhores parâmetros para a configuração do FCS, foi avaliado o desempenho do protocolo APA-MAC em relação ao estado da arte. Portanto para todas simulações a seguir, o APA-MAC com FCS fixo foi configurado com os multiplicadores 1,5x e 2 para as mensagens com 15% e 50% do ciclo respectivamente. Com o FCS variável, APA-MAC foi configurado com os multiplicadores 3x e 3,5 para as mensagens com 15% e 50% do ciclo respectivamente. Para avaliar o desempenho do protocolo, porém para um número maior de saltos, foi utilizado estes mesmos parâmetros, mantendo uma distância de 1300 metros entre a origem e o destino.

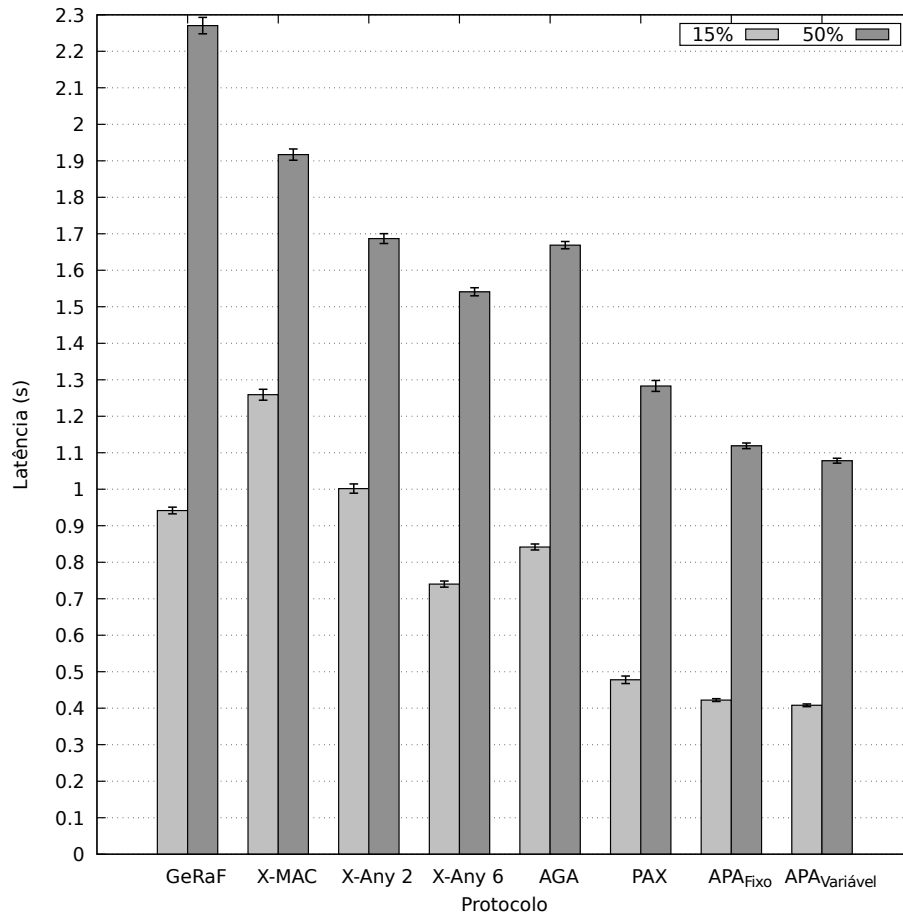
As Figuras 5.3 e 5.4 mostram a latência obtida no encaminhamento de mensagens em um cenário sem colisão, ou seja, não há fluxos concorrentes para a transmissão dos dados. Entretanto a Figura 5.3 mostra os resultados obtidos quando a distância entre origem e destino é de 650 metros. Já a Figura 5.4, mostra os resultados quando esta distância é de 1300 metros

Tanto mantendo uma distância de 650 ou 1300 metros entre a origem e o destino, a relação de desempenho entre os protocolos foram semelhantes.

O pior resultado para mensagens pequenas foi encontrado no protocolo X-MAC. Isso pode ser explicado pelo fato de que ele não usa método *anycast*, exigindo, em média 50% do ciclo a cada salto para o encaminhamento da mensagem. De acordo com estudos analíticos em Heimfarth, Giacomini e Araujo (2015), um pequeno número de nós no FCS (um no caso do X-MAC), não apresenta resultados satisfatórios para mensagens pequenas ou medianas, sendo adequado apenas para mensagens muito longas devido ao número de saltos reduzido. As cardinalidades altas para o FCS reduzem a latência quando mensagens pequenas são transmitidas enquanto que os FCS com pequeno número de nós apresentam melhor desempenho para mensagens grandes (em comparação com o tempo de ciclo).

Desta forma, o desempenho do GeRaF/CMAC também é influenciado por esta regra. Uma vez que o protocolo utiliza todos os nós que trazem melhorias de avanço até o destino (alta

Figura 5.3 – Avaliação da latência quando origem e destino estão a 650 metros

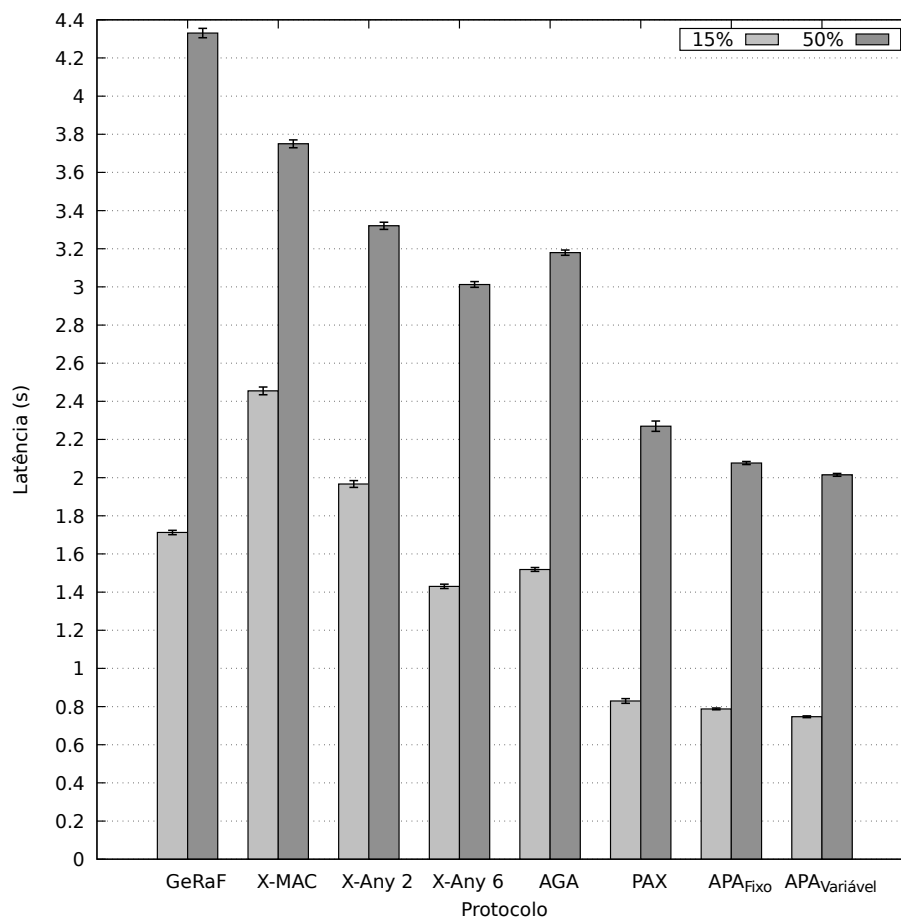


cardinalidade para as densidades utilizadas), possui um melhor desempenho para mensagens pequenas. Portanto, o resultado piora conforme o aumento do tamanho das mensagens, uma vez que a redução do tempo para encontrar o próximo transmissor, não é grande o suficiente para compensar a transmissão de mensagens longas por um número maior de saltos.

O protocolo AGA-MAC, obteve resultados inferiores ao protocolo X-MAC Anycast 6. Este resultado pode ser explicado devido a estabilidade da cardinalidade do FCS no X-MAC Anycast 6 (sempre 6), enquanto que no AGA-MAC, uma vez que a posição dos vizinhos não é conhecida, o tamanho FCS varia. Portanto no AGA-MAC, em alguns saltos, um pequeno número de nós pode fazer parte do FCS, aumentando consideravelmente o número de preâmbulos necessários para encontrar o nó do próximo salto.

O protocolo X-MAC Anycast 2, possuindo um FCS fixo igual a 2, apresentou um desempenho superior ao X-MAC, mas inferior ao AGA-MAC e ao X-MAC Anycast 6. Este resultado mostra que, para os parâmetros avaliados, uma cardinalidade do FCS próxima de 6 é mais apropriada para a classe de protocolos X-MAC Anycast.

Figura 5.4 – Avaliação da latência quando origem e destino estão a 1300 metros



Para ambos tamanhos de mensagem, o protocolo PAX-MAC obteve uma latência consideravelmente menor em relação aos outros protocolos (não incluindo APA-MAC). Isto pode ser explicado pelo fato de que preâmbulos e dados são transmitidos ao mesmo tempo em diferentes locais na rede. Uma vez que os preâmbulos avançam criando um caminho escalonado para o encaminhamento dos dados, a intenção é que a transmissão dos dados ocorra a uma velocidade constante, assim como seria em uma rede que não opera em ciclos de trabalho.

Os resultados mostraram que neste experimento, o protocolo APA-MAC, obteve o melhor desempenho no tempo de entrega das mensagens, seja utilizando um FCS fixo ou variável. Comparando ao PAX-MAC, para uma distância de 650 metros entre origem e destino, utilizando um FCS fixo, os ganhos médios obtidos utilizando um FCS fixo e um FCS variável foram respectivamente 11% e 14% para as mensagens pequenas. Para mensagens maiores, estes valores foram 12% e 15%. Mantendo uma distância de 1300 metros, as médias obtidas utilizando um FCS fixo e um FCS variável foram respectivamente 6% e 10% para as mensagens pequenas.

Para mensagens maiores, estes valores foram 8% e 11%. Dessa forma, é possível notar que o FCS variável, obteve melhor desempenho na avaliação da latência.

A redução do ganho de desempenho em relação ao PAX-MAC no encaminhamento de mensagens em maiores distâncias, pode ser explicada pelo fato que o PAX-MAC utilizando o mesmo tempo de atraso para iniciar o envio dos dados, a latência indesejável que pode ser obtida, é proporcionalmente menor se comparado a latência total. Dessa forma, somente em eventuais ocorrências de aproximação entre dados e preâmbulo, esta latência será notada.

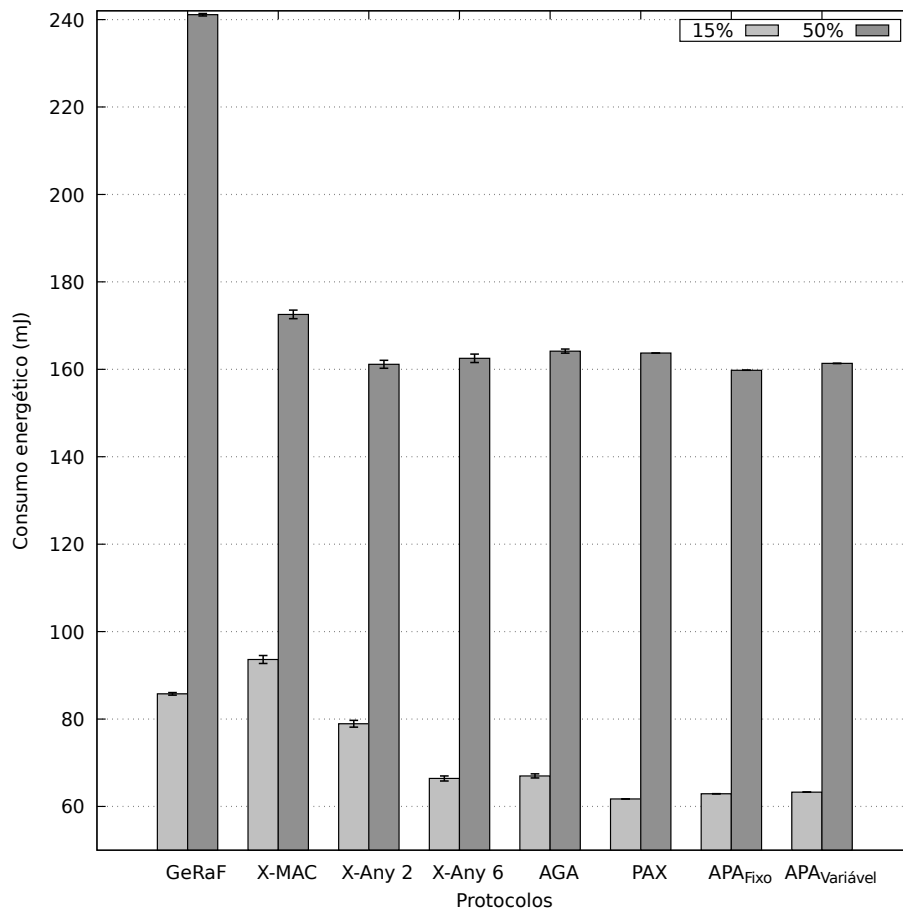
5.3 Avaliação do Consumo Energético

Para a avaliação do consumo energético, não foi considerado o consumo das operações normais da rede, como por exemplo, o período, em que os nós sensores periodicamente entram em atividade para executar o *carrier sense*. Os resultados definem apenas a energia que foi gasta para enviar a mensagem de dados da origem até o destino. Foi considerado que a escuta ociosa possui o mesmo consumo energético das transmissões. As Figuras 5.5 e 5.6 mostra os resultados para um percurso de 650 e 1300 metros respectivamente.

É possível observar em ambas as figuras que o consumo de energia apresentado pelo GeRaF/CMAC somente não foi maior que o X-MAC quando considerado as mensagens pequenas. Incluir no FCS todos os nós que podem promover o progresso na direção do destino, têm como consequência a redução no progresso médio da mensagem. Para os outros protocolos testados, o progresso médio é sempre superior ao apresentado pelo GeRaF/CMAC. Esse comportamento é explicado pelo fato de que apenas os vizinhos que promovem o progresso mais significativo fazem parte do FCS. Como consequência, mais saltos são necessários para chegar ao destino ao usar GeRaF/CMAC. Quando mensagens longas são transmitidas, o consumo de energia adicional em cada salto torna-se significativo.

Analisando os resultados de X-MAC, X-MAC Anycast 2 e X-MAC Anycast 6 em ambas figuras, é possível notar que conforme a cardinalidade do FCS aumenta, o consumo para a transmissão de mensagens pequenas diminui. Isto ocorre porque aumentando o número de elementos no FCS, menos preâmbulos são necessários para encontrar o próximo salto. Dessa forma, a economia de energia decorrente da redução das transmissões dos preâmbulos, pode ser maior que a energia gasta para os saltos acrescidos. Para mensagens maiores, X-MAC Anycast 2 e X-MAC Anycast 6 obtiveram valores próximos devido a uma troca entre menor gasto com envio de preâmbulos e maior gasto com transmissões dos dados, e vice-versa.

Figura 5.5 – Avaliação do consumo energético quando origem e destino estão a 650 metros



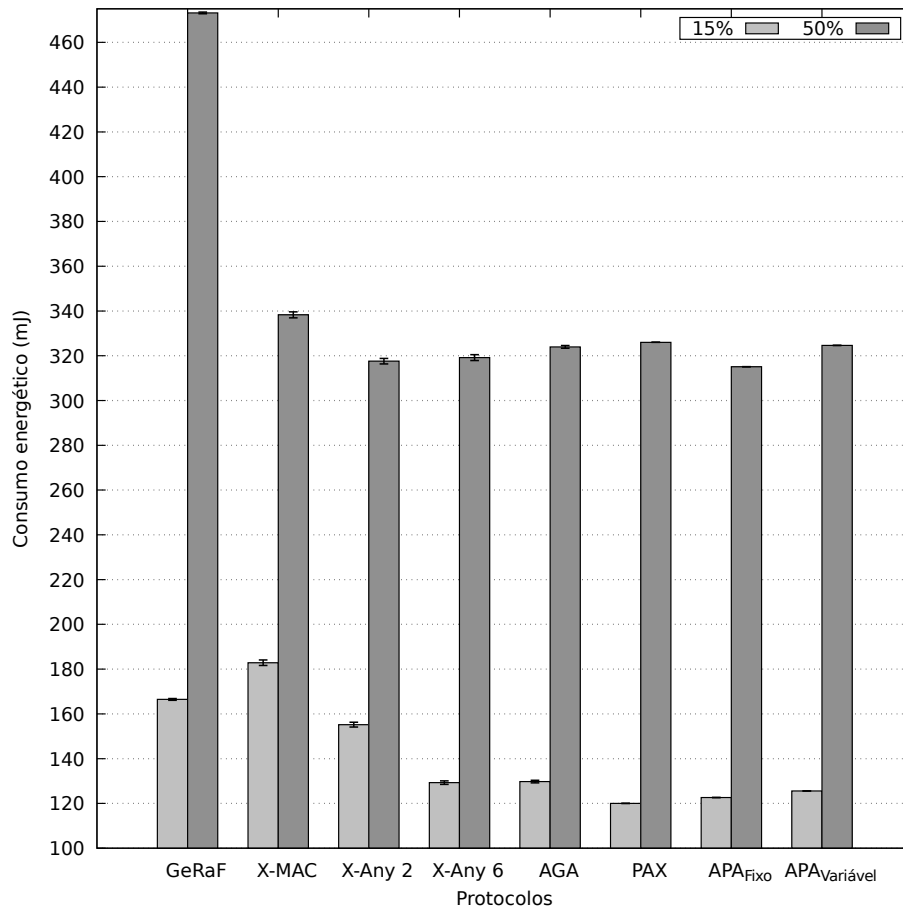
Entretanto, para altas cardinalidades do FCS, como o GeRaF/CMAC por exemplo, o consumo das transmissões de dados acrescidas, devido ao aumento de saltos, não é compensado pela redução da média preâmbulos transmitidos. Além disso, o aumento de saltos também reduz o ganho na redução dos preâmbulos, uma vez que em cada salto, também foram utilizados preâmbulos para encontrar o nó transmissor.

Para as mensagens grandes, APA-MAC com FCS fixo mostrou uma pequena vantagem em relação aos outros protocolos. Um FCS de cardinalidade 2 para mensagens com tamanho igual a 50% do ciclo, promoveu uma redução nas transmissões de preâmbulos sem um aumento relevante de saltos. Para mensagens menores, mesmo não obtendo o menor consumo, APA-MAC apresentou valores muito próximos ao do PAX-MAC.

5.4 Avaliação da Latência com Fluxos Concorrentes

Para avaliar o comportamento do APA-MAC em cenários com alto tráfego de comunicação, foram realizados experimentos com fluxos de comunicação concorrentes. Portanto, para

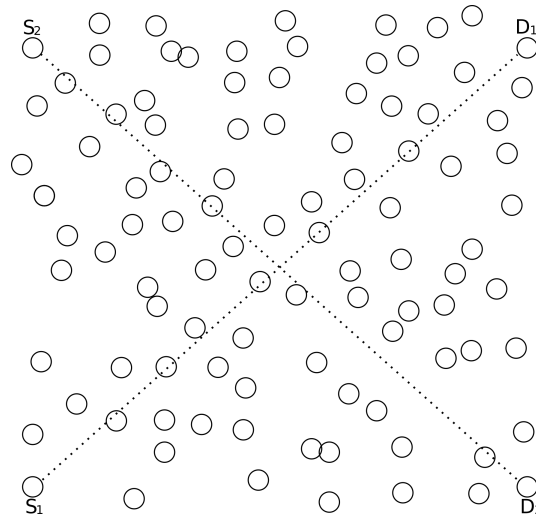
Figura 5.6 – Avaliação do consumo energético quando origem e destino estão a 1300 metros



promover a concorrência entre os fluxos de comunicação, o encaminhamento de mensagens seguiu sentidos cruzados, conforme ilustrado na Figura 5.7. Os nós $S1$ e $S2$ são as fontes de duas comunicações simultâneas diferentes e os nós $D1$ e $D2$ os respectivos destinos. Cada simulação é composta por dois fluxos de mensagem independentes, cada um iniciado por um nó de fonte (S_i) implantado num canto de uma área quadrada e fluindo para o destino correspondente (D_i) no canto oposto. Todos os nós restantes são distribuídos aleatoriamente na área. O tempo de início de cada fluxo é escolhido aleatoriamente pelo nó remetente dentro de um intervalo de 0 a 5 vezes o tempo de ciclo.

Neste experimento, foi adaptado aos protocolos avaliados, um comportamento de persistência no envio de preâmbulos (RTS). Durante a procura pelo nó do próximo salto, o nó remetente transmite preâmbulos para alcançar um vizinho ativo. Este processo se repete durante um período igual ao tempo do ciclo. Entretanto, se nenhuma resposta (CTS) é obtida, o nó emissor reinicia o envio da sequência de preâmbulos por mais um tempo do ciclo. Na

Figura 5.7 – Esquema do percurso dos fluxos concorrentes



simulação, se nenhuma resposta é obtida até o segundo reinício no envio de preâmbulos (três sequências completas), a mensagem de dados é descartada.

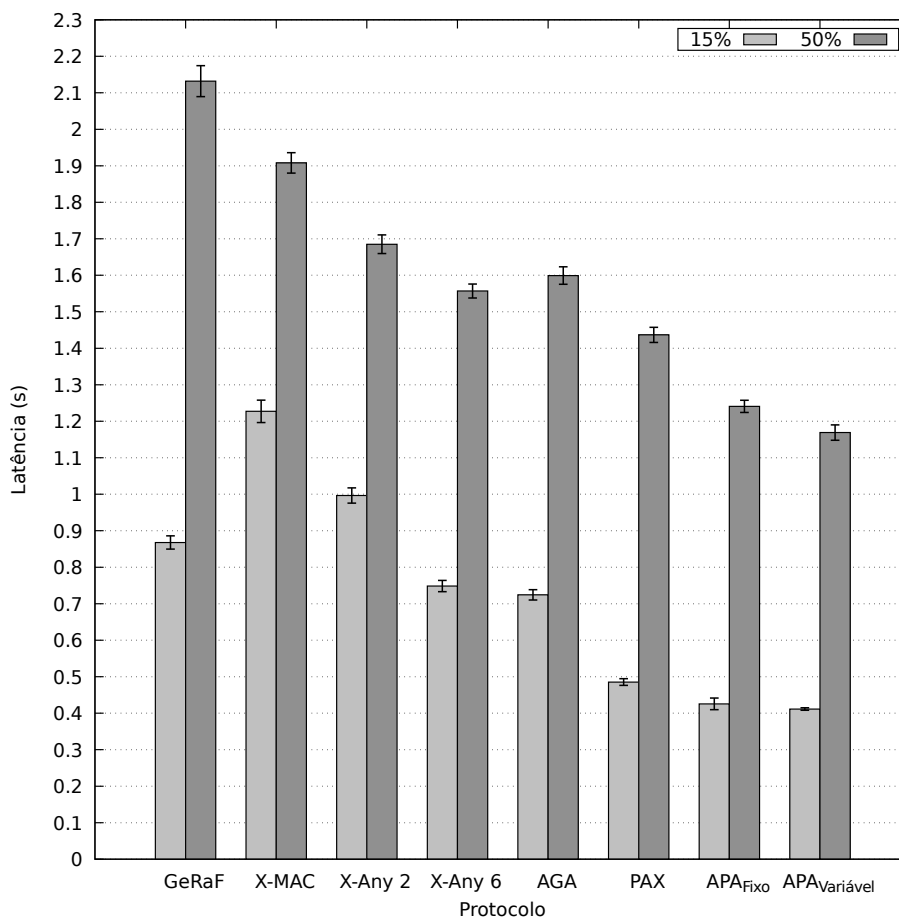
O protocolo PAX-MAC possui um comportamento ligeiramente diferente no processo de reinício de preâmbulos. Antes que as tentativas de reinício terminem, os dados irão aproximar-se da transmissão de preâmbulos. Neste caso, o nó remetente irá cessar o envio de preâmbulos e aguardar a chegada dos dados, tendo em seguida o mesmo papel do nó de origem.

A Figura 5.8 mostra a latência obtida no encaminhamento de mensagens em um cenário propício à colisão, ou seja, quando há fluxos concorrentes para a transmissão dos dados.

Como esperado, a latência média em todos os protocolos para as mensagens pequenas (15% do ciclo) foi consideravelmente menor do que para mensagens grandes. O protocolo APA-MAC obteve melhor performance em relação aos outros protocolos neste experimento. O protocolo apresentou a menor latência para ambos os tamanhos das mensagens, utilizando um FCS fixo ou variável. Para mensagens pequenas, o ganho foi próximo quando nenhum outro tráfego estava presente na rede (cenário de fluxo único).

Para mensagens grandes (50% do ciclo), os protocolos X-MAC Anycast 2, X-MAC Anycast 6 e AGA-MAC apresentam resultados semelhantes, com um desempenho pior que do PAX-MAC e APA-MAC. Os protocolos X-MAC e GeRaF/CMAC obtiveram a maior latência nesta experiência. Os resultados foram semelhantes aos obtidos no cenário de fluxo único, sendo APA-MAC, o protocolo que obteve a menor latência.

Figura 5.8 – Avaliação da latência com fluxos concorrentes



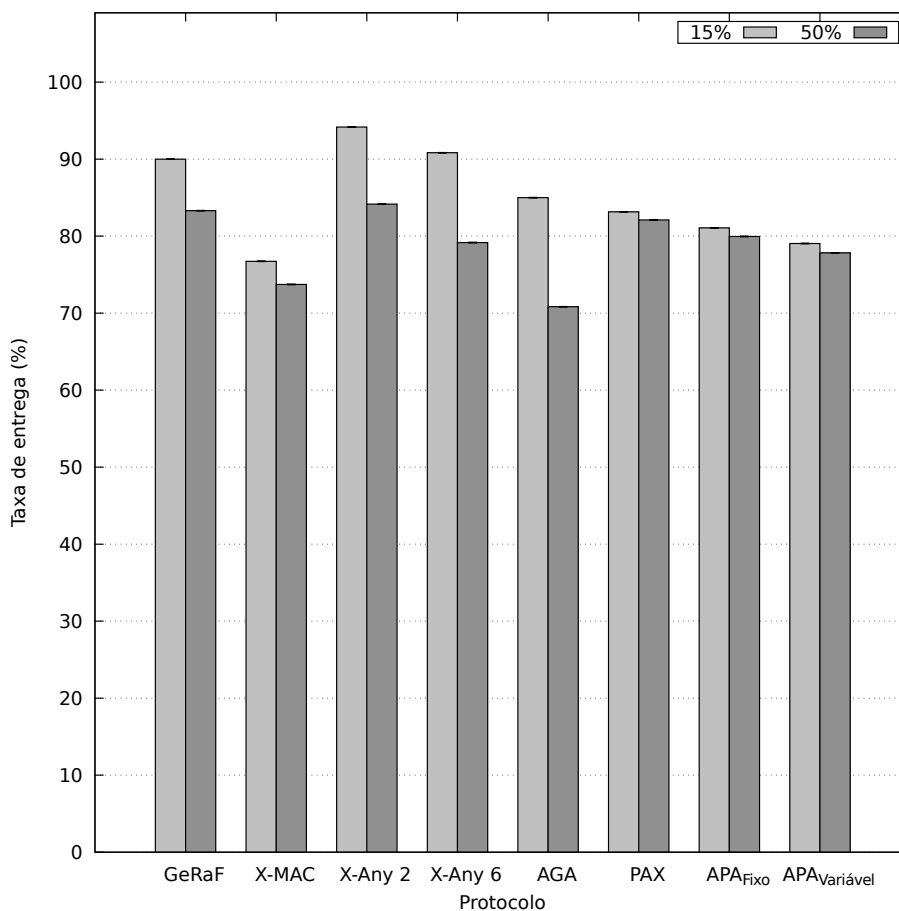
5.5 Avaliação da Taxa de Entrega com Fluxos Concorrentes

Neste trabalho também foi contabilizado a taxa de sucesso na entrega de mensagens no cenário propenso à colisões. Os resultados foram obtidos durante a execução do experimento para a avaliação da latência em fluxos concorrentes (Seção 5.4). Com exceção do APA-MAC, os protocolos avaliados não possuem mecanismos para confirmar a recepção dos dados. Caso haja uma colisão no receptor, o nó emissor não será notificado, e o encaminhamento dos dados será interrompido, ocorrendo a perda da mensagem. Para uma comparação justa entre os protocolos, neste experimento, o protocolo APA-MAC não verificou a recepção dos dados. Desta forma, imediatamente após o envio dos dados, o protocolo entra em modo inativo, voltando aos ciclos de trabalho.

A Figura 5.9 apresenta a taxa de sucesso na entrega obtida pelos protocolos avaliados, para ambos tamanhos de mensagens.

Para a maioria das configurações, para um dado protocolo, a taxa de sucesso foi maior para mensagens com duração de 15% do ciclo do que para mensagens grandes. Isso pode ser

Figura 5.9 – Avaliação da taxa de entrega com fluxos concorrentes



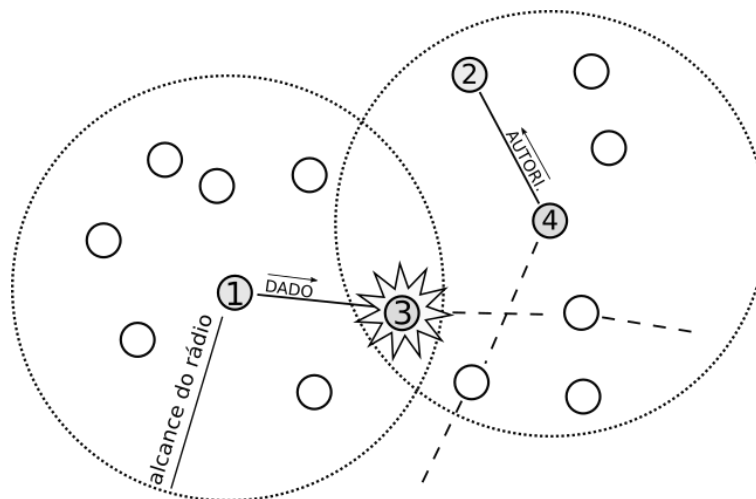
explicado pelo fato de que mensagens maiores estão usando recursos de rede por um tempo maior, aumentando a probabilidade de colisões. Seja devido a transmissão da mensagem ou o maior número de preâmbulos enviados devido ao FCS reduzido.

A baixa taxa de entrega do X-MAC pode ser explicada pelo fato de possuir apenas um elemento em seu FCS, obtendo uma menor velocidade de avanço dos preâmbulos. Desta forma, assim com na transmissão de mensagens grandes, os recursos da rede são usados por um tempo maior, aumentando a probabilidade de colisões. Entretanto, com o aumento da cardinalidade do FCS, maior a média de saltos realizados. O maior número transmissões na rede também aumenta a probabilidade de colisões. Portanto, é observado que os protocolos com FCS maiores como X-MAC Anycast 6, GeRaF/CMAC e AGA-MAC tiveram uma taxa entrega menor que X-MAC Anycast 2.

A menor taxa de entrega dos protocolos PAX-MAC e APA-MAC em relação ao protocolo X-MAC Anycast 6, pode ser explicada pelo grande intervalo de tempo de recursos bloqueados na rede, causado pela transmissão separada de preâmbulos e dados.

O protocolo APA-MAC, utiliza uma mensagem autorizando o envio dos dados somente quando o canal está livre. Entretanto na simulação foi identificado problemas de terminal oculto. A Figura 5.10, demonstra este problema.

Figura 5.10 – Terminal Oculto



Como ilustra a figura, o nó 1 está enviando uma mensagem de dados para 3. Neste momento, em outra transmissão, o nó 2 está aguardando a mensagem de autorização para enviar os dados. Uma vez que o nó 4 não escuta a transmissão de 1, identifica o canal livre e envia a mensagem de autorização para 2. Neste momento, ocorre uma colisão na recepção dos dados em 3.

Portanto, mesmo sendo um mecanismo para evitar colisões, o envio da mensagem de autorização pode ocasionar outras, já que aumenta a área de colisão. Sem a utilização de técnicas para confirmar a recepção dos dados, APA-MAC não sobressai em relação aos outros protocolos na taxa de entrega.

Com o objetivo de avaliar o desempenho do protocolo APA-MAC da forma como proposto, foi realizado outro experimento, mas dessa vez, identificando se a mensagem de dados foi entregue ao próximo salto. Neste caso, a transmissão dos dados pelo nó do próximo salto, é usado como uma confirmação de recebimento (ACK). Caso esta confirmação não seja identificada, o nó irá aguardar no próximo período outra autorização para reenviar os dados. Dessa forma, APA-MAC cria meios para se recuperar de uma colisão. Os resultados obtidos demonstraram que APA-MAC conseguiu entregar cerca de 98% das mensagens independente do tamanho, utilizando o FCS fixo ou variável.

6 Conclusões e Trabalhos Futuros

Os protocolos *anycast* foram propostos para reduzir o problema do atraso causado pelos ciclos de trabalho em redes de sensores sem fio assíncronas. Eles exploram a redundância de caminho das redes densas selecionando um grupo de nós como bons candidatos para encaminhar uma mensagem. Este grupo de nós é denominado FCS (*Forwarding Candidate Set*). No protocolo *anycast* APA-MAC, proposto neste trabalho, o nó de origem transmite uma série de preâmbulos curtos à frente para encontrar o nó do próximo salto. Todos os nós que pertencem à rota entre a fonte e o destino recebem a informação contida no preâmbulo sobre o tempo em que ocorrerá a transmissão dos dados. Dessa forma, os nós escalonam seu período de atividade para receber e encaminhar os dados. Com este processo, a transmissão simultânea de preâmbulos e dados é explorada, obtendo redução na latência.

No APA-MAC, diferente do protocolo PAX-MAC (HEIMFARTH et al., 2016), não é necessário manter um atraso inicial para evitar que a mensagem de dados se aproxime dos preâmbulos. PAX-MAC adota este mecanismo para evitar a sobrecarga de reiniciar o processo de transmissão dos dados, o que aumentaria a latência. Sem este mecanismo, APA-MAC pode iniciar o envio dos dados, assim que o receptor sinalizar que o canal está livre. Dessa forma, além de reduzir a latência, esta técnica aumenta a taxa de sucesso na entrega de mensagens.

Em Heimfarth et al. (2016), a cardinalidade do FCS é selecionada para promover a mesma velocidade de propagação de preâmbulos e dados. Entretanto, neste trabalho, a cardinalidade do FCS é selecionada para promover uma aceleração na propagação dos preâmbulos. Isto evita que o receptor encontre o canal ocupado, e ocorra uma espera para a transmissão dos dados, o que ocasionaria uma maior latência. Para evitar um grande aumento no número de saltos utilizando maiores cardinalidade do FCS, foi empregado técnicas para otimizar saltos curtos.

APA-MAC foi comparado com X-MAC, AGA-MAC, PAX-MAC e uma versão aprimorada do GeRaF baseada no CMAC. Duas versões do protocolo AnyMac também foram avaliadas, denominadas neste trabalho como X-MAC Anycast 2, X-MAC Anycast 6. As simulações demonstraram que o protocolo proposto superou todos na latência, mantendo baixo consumo energético, portanto, alcançando os objetivos deste trabalho. Em relação à latência, utilizando um FCS que adapta sua cardinalidade conforme a aproximação dos dados, APA-MAC obteve no mínimo 10% menor latência para mensagens pequenas e 11% para mensagens grandes, comparado ao segundo melhor colocado. Utilizando meios para confirmar a recepção dos dados e

se recuperar de uma colisão, o protocolo conseguiu entregar uma média de 98% das mensagens de dados, sejam elas pequenas ou grandes.

Como trabalho futuros, é pretendido aprimorar o método de redução de saltos, para que além de saltos curtos, o protocolo também possam otimizar os saltos medianos, reduzindo o número total de saltos. Também pretende-se obter modelo matemático para definir os parâmetros do FCS fixo e variável baseado na densidade da rede. Além disso, será estudada e avaliada técnicas para reduzir o número de colisões.

REFERÊNCIAS

- ABDULLAH, M.; EHSAN, A. Routing protocols for wireless sensor networks: Classifications and challenges. **Journal of Electronics and Communication Engineering Research**, v. 2, n. 2, p. 5–15, 2014.
- AKYILDIZ, I. F.; VURAN, M. C.; AKAN, O. B. A cross-layer protocol for wireless sensor networks. In: **2006 40th Annual Conference on Information Sciences and Systems**. 2006. p. 1102–1107.
- ANWANDER, M. et al. Beam: A burst-aware energy-efficient adaptive mac protocol for wireless sensor networks. In: IEEE. **Networked Sensing Systems (INSS), 2010 Seventh International Conference on**. 2010. p. 195–202.
- ASHRAF, F.; VAIDYA, N.; KRAVETS, R. Any-mac: Extending any asynchronous mac with anycast to improve delay in wsn. In: **Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2011 8th Annual IEEE Communications Society Conference on**. 2011. p. 19–27.
- AVVENUTI, M. et al. Increasing the efficiency of preamble sampling protocols for wireless sensor networks. In: IEEE. **Mobile Computing and Wireless Communication International Conference, 2006. MCWC 2006. Proceedings of the First**. 2006. p. 117–122.
- AVVENUTI, M.; VECCHIO, A. Adaptability in the b-mac+ protocol. In: IEEE. **Parallel and Distributed Processing with Applications, 2008. ISPA'08. International Symposium on**. 2008. p. 946–951.
- BUETTNER, M. et al. 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.
- CANO, C. et al. Low energy operation in wsns: A survey of preamble sampling mac protocols. **Computer Networks**, Elsevier, v. 55, n. 15, p. 3351–3363, 2011.
- CARRANO, R. C. et al. Survey and taxonomy of duty cycling mechanisms in wireless sensor networks. **IEEE Communications Surveys & Tutorials**, IEEE, v. 16, n. 1, p. 181–194, 2014.
- CASARI, P. et al. Alba: an adaptive load-balanced algorithm for geographic forwarding in wireless sensor networks. In: IEEE. **Military Communications Conference, 2006. MILCOM 2006. IEEE**. 2006. p. 1–9.
- CHENG, L. et al. Contention-based geographic forwarding in asynchronous duty-cycled wireless sensor networks. **International Journal of Communication Systems**, Wiley Online Library, v. 25, n. 12, p. 1585–1602, 2012.
- CHOUDHURY, R. R.; VAIDYA, N. H. Mac-layer anycasting in ad hoc networks. **ACM SIGCOMM Computer Communication Review**, ACM, v. 34, n. 1, p. 75–80, 2004.
- 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.

DU, S.; SAHA, A. K.; JOHNSON, D. B. Rmac: A routing-enhanced duty-cycle mac protocol for wireless sensor networks. In: IEEE. **INFOCOM 2007. 26th IEEE International Conference on Computer Communications**. IEEE. 2007. p. 1478–1486.

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.

ERGEN, S. C.; VARAIYA, P. Pedamac: Power efficient and delay aware medium access protocol for sensor networks. **IEEE Transactions on Mobile Computing**, IEEE, v. 5, n. 7, p. 920–930, 2006.

FARAHANI, S. **ZigBee wireless networks and transceivers**. 1. ed. : newnes, 2011. ISBN 0750683937, 978-0750683937.

FIELDCOMM GROUP. **Hart Technology**. 2017. Disponível em: <<https://fieldcommgroup.org/technologies/hart/hart-technology>>. Acesso em: 17 abr. 2017.

HEIMFARTH, T.; GIACOMIN, J.; ARAUJO, J. D. Aga-mac: Adaptive geographic anycast mac protocol for wireless sensor networks. In: **Advanced Information Networking and Applications (AINA), 2015 IEEE 29th International Conference on**. 2015. p. 373–381.

HEIMFARTH, T. et al. A preamble ahead anycast protocol for wsns. In: IEEE. **Advanced Information Networking and Applications (AINA), 2016 IEEE 30th International Conference on**. 2016. p. 458–466.

HOESEL, L. F. V.; HAVINGA, P. J. A lightweight medium access protocol (lmac) for wireless sensor networks: Reducing preamble transmissions and transceiver state switches. In: **1st International Workshop on Networked Sensing Systems (INSS)**. : Society of Instrument and Control Engineers (SICE), 2004.

HONG, S.-H.; KIM, H.-k. A multi-hop reservation method for end-to-end latency performance improvement in asynchronous mac-based wireless sensor networks. **IEEE Transactions on Consumer Electronics**, IEEE, v. 55, n. 3, 2009.

HUANG, P. et al. The evolution of mac protocols in wireless sensor networks: A survey. **IEEE communications surveys & tutorials**, IEEE, v. 15, n. 1, p. 101–120, 2013.

HURNI, P.; BRAUN, T. Maxmac: A maximally traffic-adaptive mac protocol for wireless sensor networks. In: SPRINGER. **European Conference on Wireless Sensor Networks**. 2010. p. 289–305.

IEEE STANDARDS ASSOCIATION. **Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications**. 2012. Disponível em: <<http://standards.ieee.org/getieee802/download/802.11-2012.pdf>>. Acesso em: 17 abr. 2017.

IEEE STANDARDS ASSOCIATION. **Part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks**. 2015. Disponível em: <<http://standards.ieee.org/getieee802/download/802.15.4-2015.pdf>>. Acesso em: 17 abr. 2017.

- INTERNATIONAL SOCIETY OF AUTOMATION. **Wireless Systems For Industrial Automation: Process Control And Related Applications**. 2017. Disponível em: <<https://infostore.saiglobal.com/store/Details.aspx?productID=1504748>>. Acesso em: 17 abr. 2017.
- JURDAK, R.; BALDI, P.; LOPES, C. V. Energy-aware adaptive low power listening for sensor networks. In: **Proceedings of INSS**. 2005. p. 24–29.
- KARP, B.; KUNG, H. T. Gpsr: greedy perimeter stateless routing for wireless networks. In: **Proceedings of the 6th annual international conference on Mobile computing and networking**. New York, NY, USA: ACM, 2000. (MobiCom '00), p. 243–254. ISBN 1-58113-197-6.
- KECHAR, B. et al. Energy efficient cross-layer mac protocol for wireless sensor networks. In: **Proceedings of the second international conference on verification and evaluation of computer and communication systems (VECoS'08)**. 2008. p. 61–71.
- KIM, J. et al. Minimizing delay and maximizing lifetime for wireless sensor networks with anycast. **IEEE/ACM Transactions on Networking (TON)**, IEEE Press, v. 18, n. 2, p. 515–528, 2010.
- KREDO, K.; MOHAPATRA, P. Medium access control in wireless sensor networks. **Computer networks**, Elsevier, v. 51, n. 4, p. 961–994, 2007.
- LESSMANN, J.; HEIMFARTH, T.; JANACIK, P. Shox: An easy to use simulation platform for wireless networks. In: IEEE. **Computer Modeling and Simulation, 2008. UKSIM 2008. Tenth International Conference on**. 2008. p. 410–415.
- LI, J. et al. Connectivity, coverage and placement in wireless sensor networks. **Sensors**, v. 9, n. 10, p. 7664, 2009. ISSN 1424-8220.
- LI, Z.; LI, M.; LIU, Y. Towards energy-fairness in asynchronous duty-cycling sensor networks. **ACM Transactions on Sensor Networks (TOSN)**, ACM, v. 10, n. 3, p. 38, 2014.
- LIM, S. et al. Medium access control with an energy-efficient algorithm for wireless sensor networks. In: SPRINGER. **IFIP International Conference on Personal Wireless Communications**. 2006. p. 334–343.
- LIN, E.-Y.; RABAEY, J. M.; WOLISZ, A. Power-efficient rendez-vous schemes for dense wireless sensor networks. In: IEEE. **Communications, 2004 IEEE International Conference on**. 2004. v. 7, p. 3769–3776.
- LIU, B.; ZHANG, L.; ZHANG, H. An adaptive schedule medium access control for wireless sensor networks. In: IEEE. **Networking, 2007. ICN'07. Sixth International Conference on**. 2007. p. 12–12.
- LIU, S.; FAN, K.-W.; SINHA, P. Cmac: An energy-efficient mac layer protocol using convergent packet forwarding for wireless sensor networks. **ACM Transactions on Sensor Networks (TOSN)**, ACM, v. 5, n. 4, p. 29, 2009.
- 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.

MENDES, L. D.; RODRIGUES, J. J. A survey on cross-layer solutions for wireless sensor networks. **Journal of Network and Computer Applications**, Elsevier, v. 34, n. 2, p. 523–534, 2011.

MERLIN, C. J.; HEINZELMAN, W. B. Network-aware adaptation of mac scheduling for wireless sensor networks. In: CITESEER. **Proc. of the Conference on Distributed Computing in Sensor Systems (Poster Session)**. 2007. p. 24–28.

MICROCHIP. **Low-cost Embedded Wireless Connectivity for Commercial and Smart Home Networks**. 2017. Disponível em: <<http://www.microchip.com/design-centers/wireless-connectivity/embedded-wireless/802-15-4/miwi-protocol>>. Acesso em: 17 abr. 2017.

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.

RAULT, T.; BOUABDALLAH, A.; CHALLAL, Y. Energy efficiency in wireless sensor networks: A top-down survey. **Computer Networks**, Elsevier, v. 67, p. 104–122, 2014.

RAWAT, P. et al. Wireless sensor networks: a survey on recent developments and potential synergies. **The Journal of supercomputing**, Springer, v. 68, n. 1, p. 1–48, 2014.

SANCHEZ, E. R.; CHAUDET, C.; REBAUDENGO, M. Improving preamble sampling performance in wireless sensor networks with state information. In: IEEE. **Wireless On-Demand Network Systems and Services, 2009. WONS 2009. Sixth International Conference on**. 2009. p. 101–108.

SHELBY, Z.; BORMANN, C. **6LoWPAN: The wireless embedded Internet**. 1. ed. : John Wiley & Sons, 2011. v. 43.

STONE, K.; COLAGROSSO, M. Efficient duty cycling through prediction and sampling in wireless sensor networks. **Wireless Communications and Mobile Computing**, Wiley Online Library, v. 7, n. 9, p. 1087–1102, 2007.

SUN, Y. et al. Dw-mac: a low latency, energy efficient demand-wakeup mac protocol for wireless sensor networks. In: ACM. **Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing**. 2008. p. 53–62.

SUN, Y.; GUREWITZ, O.; JOHNSON, D. B. Ri-mac: a receiver-initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks. In: ACM. **Proceedings of the 6th ACM conference on Embedded network sensor systems**. 2008. p. 1–14.

TAN, H.-X.; CHAN, M. C. A²-mac: An adaptive, anycast mac protocol for wireless sensor networks. In: IEEE. **Wireless Communications and Networking Conference (WCNC), 2010 IEEE**. 2010. p. 1–6.

TOKLU, S.; ERDEM, O. A. Bsc-mac: energy efficiency in wireless sensor networks with base station control. **Computer Networks**, Elsevier, v. 59, p. 91–100, 2014.

UNIVERSIDADE FEDERAL DE LAVRAS. Biblioteca Universitária. **Manual de normalização e estrutura de trabalhos acadêmicos: TCCs, monografias, dissertações e teses**. 2. ed. Lavras, 2017. Disponível em: <<http://repositorio.ufla.br/jspui/handle/1/11017>>. Acesso em: 17 abr. 2017.

VURAN, M. C.; AKYILDIZ, I. F. Xlp: A cross-layer protocol for efficient communication in wireless sensor networks. **IEEE transactions on mobile computing**, IEEE, v. 9, n. 11, p. 1578–1591, 2010.

WONG, K.-J.; ARVIND, D. Speckmac: low-power decentralised mac protocols for low data rate transmissions in specknets. In: ACM. **Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality**. 2006. p. 71–78.

YANG, S.-H. **Wireless Sensor Networks: Principles, Design and Applications**. 1. ed. : Springer Publishing Company, Incorporated, 2013. ISBN 1447155041, 9781447155041.

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. IEEE**. 2002. v. 3, p. 1567–1576.

YE, W.; SILVA, F.; HEIDEMANN, J. Ultra-low duty cycle mac with scheduled channel polling. In: ACM. **Proceedings of the 4th international conference on Embedded networked sensor systems**. 2006. p. 321–334.

ZENG, K. et al. On geographic collaborative forwarding in wireless ad hoc and sensor networks. In: IEEE. **Wireless Algorithms, Systems and Applications, 2007. WASA 2007. International Conference on**. 2007. p. 11–18.

ZIN, S. M. et al. Routing protocol design for secure wsn: Review and open research issues. **Journal of Network and Computer Applications**, Elsevier, v. 41, p. 517–530, 2014.

ZORZI, M.; RAO, R. R. Geographic random forwarding (geraf) for ad hoc and sensor networks: multihop performance. **Mobile Computing, IEEE Transactions on**, IEEE, v. 2, n. 4, p. 337–348, 2003.

ZUO, J. et al. Cross-layer aided energy-efficient opportunistic routing in ad hoc networks. **IEEE Transactions on Communications**, IEEE, v. 62, n. 2, p. 522–535, 2014.