



GUSTAVO CARVALHO SOUZA

**MAC-AHEAD: UM PROTOCOLO CROSS-LAYER ANYCAST
ASSÍNCRONO ADAPTÁVEL ÀS CONDIÇÕES DO CANAL DE
COMUNICAÇÃO**

LAVRAS – MG

2019

GUSTAVO CARVALHO SOUZA

**MAC-AHEAD: UM PROTOCOLO CROSS-LAYER ANYCAST ASSÍNCRONO
ADAPTÁVEL ÀS CONDIÇÕES DO CANAL DE COMUNICAÇÃO**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para a obtenção do título de Mestre.

Prof. DSc. Tales Heimfarth

Orientador

Prof. DSc. João Carlos Giacomini

Coorientador

**LAVRAS – MG
2019**

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

Souza, Gustavo Carvalho

MAC-AHEAD: um protocolo cross-layer anycast assíncrono adaptável às condições do canal de comunicação / Gustavo Carvalho Souza. - 2019.

78 p. : il.

Orientador(a): Tales Heimfarth.

Coorientador(a): João Carlos Giacomin.

Dissertação (mestrado acadêmico) - Universidade Federal de Lavras, 2019.

Bibliografia.

1. Protocolo MAC. 2. Erros no Canal de Comunicação. 3. Redes de Sensores Sem Fio. I. Heimfarth, Tales. II. Giacomin, João Carlos. III. Título.

GUSTAVO CARVALHO SOUZA

**MAC-AHEAD: UM PROTOCOLO CROSS-LAYER ANYCAST ASSÍNCRONO
ADAPTÁVEL ÀS CONDIÇÕES DO CANAL DE COMUNICAÇÃO**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para a obtenção do título de Mestre.

APROVADA em 28 de Fevereiro de 2019

Prof. Dr. Tales Heimfarth DCC/UFLA
Prof. Dr. João Carlos Giacomini DCC/UFLA
Prof. Dr. Luiz Henrique Andrade Correia DCC/UFLA
Prof. Dr. Luiz Filipe Menezes DCC/UFMG

Prof. DSc. Tales Heimfarth
Orientador

Prof. DSc. João Carlos Giacomini
Co-Orientador

**LAVRAS – MG
2019**

Dedicado a Deus e a minha Família

AGRADECIMENTOS

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pela bolsa a mim concedida para a realização deste projeto.

À Universidade Federal de Lavras especialmente ao Departamento de Ciência da Computação, pela oportunidade.

Ao orientador Tales Heimfarth e ao co-orientador João Carlos Giacomini pelas orientações para a concretização deste trabalho.

À minha querida esposa Gislene, pelo companheirismo, amor e apoio em todos momentos.

Às minhas filhas Helena e Cecília por serem minhas fontes de inspiração para a construção de novos projetos.

Ao Grande Arquiteto do Universo que me concede motivação e energia para a escalada em níveis mais altos.

RESUMO

Este trabalho apresenta o MAC-AHEAD, um protocolo cross-layer anycast assíncrono para redes de sensores sem fio (RSSFs). Este protocolo utiliza ciclo de trabalho inativo/ativo como forma de diminuir o consumo de energia e emprega o modelo de comunicação anycast para reduzir o tempo de propagação de mensagens de dados. Um método de envio de mensagens dividido em fases é responsável pela redução de latência na entrega de dados. Na primeira fase, uma sequência de mensagens de controle RTS é enviada para descobrir o próximo salto na rota e escalonar seu tempo de despertar. Na segunda fase, a mensagem DATA é enviada alguns saltos após o RTS. Assim, o RTS e o DATA percorrem a rota de origem para o destino simultaneamente, reduzindo a latência, sem consumo de energia extra. Para acelerar a propagação do RTS, um método anycast é empregado. Um conjunto de vizinhos, denominado FCS (Forward Candidate Set), são escolhidos como bons encaminhadores para a mensagem de dados. O protocolo MAC-AHEAD seleciona como próximo roteador o membro do FCS que primeiro responde ao RTS. A seleção dos membros do FCS é baseada em dois parâmetros, a proximidade do nó Destino e a SINR (relação sinal-ruído/interferência) percebido pelo nó vizinho ao receber a mensagem RTS. A fim de avaliar o desempenho do MAC-AHEAD, este foi comparado com os protocolos X-MAC e PAX-MAC. Os resultados das simulações mostraram, para os cenários testados, que o protocolo proposto supera os outros dois em latência e consumo de energia.

Palavras-chave: Protocolo MAC, Erros no Canal de Comunicação, Redes de Sensores Sem Fio

ABSTRACT

This work presents the MAC-AHEAD, an asynchronous cross-layer anycast protocol for Wireless Sensor Networks (WSNs). This protocol uses inactive/active duty cycle as a way to reduce power consumption and employs the anycast communication model to reduce the propagation time of data messages. A phased messaging method is responsible for reducing latency in data delivery. In the first phase, a sequence of RTS control messages is sent to discover the next hop on the route and to stagger its wake-up time. In the second phase, the DATA message is sent some hops after the RTS. Thus, RTS and DATA roam the route from source to destination simultaneously, reducing latency without extra power consumption. To speed up the propagation of RTS, an anycast method is employed. A set of neighbors, called FCS (Forward Candidate Set), are chosen as good forwarders for the data message. The MAC-AHEAD protocol selects as the next router the FCS member that first responds to the RTS. The selection of FCS members is based on two parameters, the proximity of the destination node and the SINR (signal-to-noise/interference-ratio) perceived by the neighboring node when receiving the RTS message. In order to evaluate the performance of MAC-AHEAD, this was compared with the X-MAC and PAX-MAC protocols. The results of the simulations showed, for the scenarios tested, that the proposed protocol surpasses the other two in latency and energy consumption.

Keywords: MAC Protocols, PAX-MAC, Communication Channel Errors, Wireless Sensor Networks.

LISTA DE FIGURAS

Figura 2.1 – Componentes de Nós Sensores	21
Figura 2.2 – Abordagem TDMA como protocolo MAC	23
Figura 2.3 – Protocolo com períodos Ativos Comuns	24
Figura 2.4 – Protocolo MAC Assíncrono (Amostragem de Preâmbulos)	25
Figura 2.5 – Disseminação de mensagens. Nós com cores pretas são os nós transmissores da mensagem.	28
Figura 3.1 – Exemplo de encaminhamento no modo guloso. O nó x encaminha dado para o nó y.	36
Figura 3.2 – Nó x realizando o roteamento no modo perímetro para o nó de destino final D.	36
Figura 4.1 – Visão geral do protocolo MAC-AHEAD.	40
Figura 4.2 – Máquina de estado do protocolo MAC-AHEAD.	41
Figura 4.3 – Formato do pacote RTS.	44
Figura 4.4 – Seleção de nós do FCS a partir de um mínimo local (nó 4). Nós 5 e 7 são selecionados por estarem dentro de um intervalo angular α	49
Figura 4.5 – Cenário apresentando as colisões entre mensagens de dados (direção pra baixo) e mensagens RTS (direção pra cima) ocorridas ao contornar buracos.	50
Figura 6.1 – Latências obtidas na transmissão do dado com tamanho de 15% do tempo de ciclo, para diferentes tempos de espera e valores de Reqs. O ponto destaca a combinação que permitiu que o pacote fosse entregue com a menor latência.	56
Figura 6.2 – Latências obtidas na transmissão do dado com tamanho de 20% do tempo de ciclo. para diferentes tempos de espera e Reqs. O ponto destaca a combinação que permitiu que o pacote fosse entregue com a menor latência.	57
Figura 6.3 – Latências obtidas na transmissão do dado com tamanho de 25% do tempo de ciclo. para diferentes tempos de espera e Reqs. O ponto destaca a combinação que permitiu que o pacote fosse entregue com a menor latência.	58
Figura 6.4 – Latências obtidas na transmissão do dado com tamanho de 33% do tempo de ciclo. para diferentes tempos de espera e valores de Reqs. O ponto destaca a combinação que permitiu que o pacote fosse entregue com a menor latência.	59

Figura 6.5 – Latências obtidas na transmissão do dado com tamanho de 50% do tempo de ciclo. para diferentes tempos de espera e valores de Reqs. O ponto destaca a combinação que permitiu que o pacote fosse entregue com a menor latência.	60
Figura 6.6 – Latências obtidas na transmissão de diferentes tamanhos de pacote de dado, para os protocolos MAC-AHEAD e PAX-MAC. MAC-AHEAD utiliza um tamanho de FCS variável	61
Figura 6.7 – Latências obtidas na transmissão de diferentes tamanhos de pacote de dado.	63
Figura 6.8 – Número de hops utilizados para a transmissão de diferentes tamanhos de pacotes de dado.	64
Figura 6.9 – Média de Perda de dados por transmissão de diferentes tamanhos de pacote de dado.	65
Figura 6.10 – Porcentagem média de colisões ao longo da transmissão de diferentes tamanhos de pacotes de dado.	66
Figura 6.11 – Consumos energéticos resultantes da transmissão de diferentes tamanhos de pacote de dado.	66
Figura 6.12 – Cenário 1 apresentando um buraco circular no terreno. A região de cor cinza representa a região onde os outros nós foram espalhados e a região de cor branca representa a região do buraco.	67
Figura 6.13 – Cenário 2 apresentando um buraco no terreno em formato da letra U. A região de cor cinza representa a região onde os outros nós foram espalhados e a região de cor branca representa a região do buraco.	68
Figura 6.14 – Latência resultante da transmissão de pacotes de dados de tamanhos diferentes para cada um dos três protocolos no cenário 1.	69
Figura 6.15 – Consumo energético resultante da transmissão de pacotes de dado de tamanhos diferentes para cada um dos três protocolos no cenário 1.	70
Figura 6.16 – Taxa de entrega dos pacotes de dados transmitidos, utilizando o cenário 1.	71
Figura 6.17 – Latência resultante da transmissão de pacotes de dados de tamanhos diferentes para cada um dos dois protocolos no cenário 2.	72
Figura 6.18 – Consumo energético resultante da transmissão de pacotes de dados de tamanhos diferentes para cada um dos dois protocolos no cenário 2.	73

LISTA DE TABELAS

Tabela 3.1 – Cabeçalho do pacote GPSR (KARP; KUNG, 2000)	37
Tabela 4.1 – Descrição dos estados utilizados na máquina de estado	42
Tabela 4.2 – Descrição das transições utilizadas na máquina de estado	43
Tabela 4.3 – Latência obtida na entrega de dados de tamanho de 15% do tempo de ciclo com um Req de 28 m. A menor latência deu-se com um ângulo de 10°	48
Tabela 4.4 – Características dos protocolos MAC-AHEAD e PAX-MAC	52
Tabela 5.1 – Parâmetros utilizados na simulação experimental	54

LISTA DE QUADROS

Quadro 6.1 – Parâmetros que resultaram em uma menor latência para o protocolo MAC-AHEAD quando realizada a transmissão do dado.	55
Quadro 6.2 – Parâmetros que permitiram ao protocolo ter a menor taxa de erros de pacotes durante a transmissão do dado.	61
Quadro 6.3 – Diferença entre a maior e a menor latência para a entrega do dado pelo protocolo X-MAC e MAC-AHEAD, para os diferentes tamanhos de pacote.	63

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Motivação	14
1.2	Definição do Problema	14
1.3	Objetivos	15
1.4	Solução Proposta	16
1.5	Estrutura do Trabalho	16
2	REVISÃO TEÓRICA	18
2.1	Redes de sensores sem fio	18
2.1.1	Arquitetura de comunicação de RSSF	18
2.1.2	Projeto de RSSF: fatores	20
2.2	Camada MAC em RSSF	22
2.3	Tipos de protocolos MAC para RSSF	23
2.4	Camada de rede em RSSF	25
2.4.1	Tipos de protocolos da camada de rede em RSSF	26
2.4.2	Disseminação de Mensagens	28
2.5	Projeto <i>cross-layer</i>	29
2.6	Controle de erros em RSSF	29
2.7	Modelo de referência para RSSFs IEEE 802.15.4	31
3	TRABALHOS RELACIONADOS	32
3.1	Protocolos	32
3.2	PAX-MAC: Protocolo assíncrono e anycast de preâmbulos antecipados para RSSF	33
3.2.1	Seleção do FCS	34
3.2.2	Cardinalidade do FCS	35
3.3	<i>Greed Perimeter Stateless Routing (GPSR)</i>	35
3.3.1	Modo guloso (<i>greedy</i>)	35
3.3.2	Modo perímetro (<i>perimeter</i>)	36
3.3.3	Execução do protocolo GPSR	37
4	Protocolo MAC-AHEAD	39
4.1	Visão geral	39
4.2	Máquina de estado	41

4.3	Pacotes RTS	44
4.4	Seleção de nós do FCS	45
4.5	Heurística para determinar a cardinalidade do FCS	45
4.6	Protocolo MAC-AHEAD: roteamento no modo perímetro para comunicação <i>anycast</i> com preâmbulos avançados	46
4.6.1	Fluxo de pacotes RTS: roteamento	46
4.6.2	Algoritmo para a escolha de nós utilizando comunicação <i>anycast</i> em cenários com buraco	47
4.6.3	Problemas no desempenho do MAC-AHEAD ao se utilizar roteamento no modo perímetro	49
4.7	Resumo das características dos protocolos MAC-AHEAD e PAX-MAC . . .	52
5	Configuração dos experimentos	53
6	Resultados e discussões	55
6.1	Definição do melhor tempo de espera e Req aceitável	55
6.2	Heurística para escolha do número de nós do FCS - Desempenho	60
6.3	Protocolos MAC-AHEAD, X-MAC e PAX-MAC - Cenário sem buracos . . .	62
6.3.1	Análise comparativa da Latência	62
6.3.2	Análise comparativa do consumo energético	65
6.4	Procolos MAC-AHEAD, X-MAC e PAX-MAC - Cenário com buracos	67
6.4.1	Cenário 1 - Análise comparativa	68
6.4.2	Cenário 2 - Análise comparativa	71
7	CONSIDERAÇÕES FINAIS	74
	REFERÊNCIAS	75

1 INTRODUÇÃO

As Redes de Sensores Sem Fio (RSSF) são redes caracterizadas por apresentarem um grande número de nós sensores distribuídos, sendo utilizadas principalmente para o monitoramento de ambientes. Alguns exemplos de setores que utilizam RSSFs são: militares, industriais, de aviação, de engenharia, de tráfego dentre outras.

Dada esta característica de monitoramento, as RSSFs apresentam restrições de energia e capacidade de auto-configuração e gerenciamento, para possíveis perdas de comunicação e falhas em nós sensores. Sendo assim, os protocolos utilizados em RSSFs são diferentes dos protocolos utilizados em redes tradicionais.

Como os nós sensores possuem capacidade energética restrita, as técnicas de *duty-cycle* são empregadas na camada de controle de acesso ao meio (MAC), mantendo cada nó em um estado de baixa energia a maior parte do tempo (> 90%). Periodicamente, cada nó sensor entra no modo ativo para realizar atividades de sensoriamento e comunicação. Esse método de operação leva a um atraso em cada transmissão de mensagem, pois o nó emissor deve aguardar o período ativo do próximo salto. Esse fenômeno é denominado *sleep-delay*. Para lidar com este problema a estratégia mais eficaz é escalonar cuidadosamente os tempos em que os nós estarão ativos ao longo do caminho em direção ao destino. Este método foi explorado pelos protocolos RMAC (DU; SAHA; JOHNSON, 2007), DMAC (LU; KRISHNAMACHARI; RAGHAVENDRA, 2004) e EX-MAC (HONG; KIM, 2009).

Os protocolos assíncronos são mais eficientes em termos de energia em RSSFs de baixo tráfego. A estratégia básica para estabelecer a comunicação dos dados é enviar longos preâmbulos antes dos próprios dados. A fim de acelerar o fluxo de dados, um método de comunicação *anycast* pode ser empregado (ASHRAF; VAIDYA; KRAVETS, 2011; KIM et al., 2010; TAN; CHAN, 2010; ZENG et al., 2007). Os protocolos *anycast* existentes normalmente unificam a funcionalidade dos protocolos MAC assíncronos, como X-MAC (BUETTNER et al., 2006) e Wise-MAC (EL-HOYDI; DECOTIGNIE, 2004), com protocolos de roteamento oportunistas (ZENG et al., 2007). Na comunicação *anycast*, um remetente seleciona um conjunto de nós sensores candidatos, compondo o FCS (*Forwarding Candidate Set*), do qual apenas um nó será escolhido para encaminhar a mensagem. O primeiro nó do FCS a responder um preâmbulo torna-se o próximo nó.

O protocolo MAC-AHEAD é um protocolo *cross-layer* assíncrono *anycast* dividido em duas fases. Na primeira fase, uma sequência de sinais de controle RTS (*request to send*) é

enviada como preâmbulos. O primeiro membro do FCS que entra no período ativo responde ao remetente e assume o papel do próximo salto. Imediatamente, ele começa a enviar preâmbulos repetindo a primeira fase. A transmissão dos dados não ocorre imediatamente após os preâmbulos. O nó emissor aguarda por um período de tempo para garantir que não ocorra colisão entre os pacotes de dados e RTS. Consequentemente, os preâmbulos e o pacote de dados propagam-se ao mesmo tempo em diferentes regiões da rede. Os preâmbulos transportam o horário da recepção do pacote de dados, o que ocorre momentos depois. Após o preâmbulo, os nós sensores são programados para acordar em um modo escalonado para encaminhar o pacote de dados sem atraso. A transmissão simultânea de pacotes de dados e sinais de controle é um mecanismo eficiente para reduzir a latência que não é encontrada em outros protocolos assíncronos anycast de última geração, como o C-MAC (LIU; FAN; SINHA, 2007) e Any-MAC (ASHRAF; VAIDYA; KRAVETS, 2011). Além disso, o MAC-AHEAD considera o tamanho dos dados para determinar a cardinalidade do FCS.

O MAC-AHEAD estima o estado atual do canal em relação à probabilidade de erros de transmissão de dados (*Packet Error Rate* - PER) devido a interferência e ruído. Este parâmetro combinado com informações de roteamento geográfico são empregados para selecionar os nós do FCS.

MAC-AHEAD é uma evolução do PAX-MAC (HEIMFARTH et al., 2016), outro protocolo *anycast* dividido em fases. As melhorias do MAC-AHEAD são: um mecanismo para lidar com ambientes com interferência de sinal e ruídos, um método para lidar com terrenos que possuem buracos e a utilização do número de elementos do FCS variável, ausentes no protocolo PAX-MAC. O termo buraco se refere a regiões da RSSF sem a presença de nós sensores. Esses espaçamentos apresentam tamanhos maiores do que o alcance de transmissão do rádio e inviabilizam a entrega do dado, caso se utilize algoritmos de roteamento que escolham apenas nós que apresentem maior avanço.

1.1 Motivação

As RSSFs são utilizadas para monitoramento e vigilância de ambientes, sendo que em grande parte das vezes, são projetadas para realizar o monitoramento de maneira autônoma e sem intervenção humana. Sendo assim, é importante que as RSSFs tenham uma vida útil prolongada com o mínimo de desperdício de energia possível.

Para economizar energia, os nós sensores utilizam ciclos de trabalho, permanecendo em estado de baixa energia a maior parte do tempo (>90%). Esta forma de operação causa um grande aumento na latência para a entrega de mensagens ao nó destino. É preciso que se encontrem estratégias de redução da latência em redes que utilizam ciclos de trabalho. Além disso, a presença de ruído e interferência nos sinais de comunicação de rádio causam erros na decodificação de mensagens. Nesses casos, a retransmissão de mensagens causa um aumento na latência e no gasto de energia.

Sendo assim, estratégias para reduzir a perda de mensagens também devem ser estudadas para aumentar a eficiência das RSSFs.

A falta de nós sensores em uma região de monitoramento que utiliza RSSFs pode ser criada durante a aplicação dos nós ao terreno ou durante o funcionamento da rede, ocasionado por nós sensores que se desgastaram energeticamente ou por outros motivos. Protocolos MAC com a capacidade de contornar estas irregularidades possuem maior autonomia e poder de operação, visto que não terá o funcionamento da rede nem a entrega de dados comprometida.

1.2 Definição do Problema

Muitas técnicas tem sido utilizada para a redução da latência e do consumo de energia em protocolos MAC para RSSF. Para se reduzir o consumo de energia, utilizam-se ciclos de trabalho, alternando o estado dos nós sensores em ativo/inativo durante o funcionamento da rede, deixando os nós em estado de inatividade na maior parte do tempo (>90%). O uso de ciclos de trabalho causa um atraso na propagação de mensagens pela rede, uma vez que um nó transmissor tenha que aguardar até que o receptor entre no estado ativo para estabelecer comunicação.

Uma das formas para se reduzir a latência causada pelo uso de ciclos de trabalho é o emprego de protocolos de comunicação do tipo anycast, em que um grupo de nós sensores é escolhido, a cada salto, como candidatos a retransmissores de mensagens.

Outra fonte de atraso na propagação de mensagens e de desperdício de energia são os erros de comunicação devido a ruído e interferência no canal. Uma mensagem recebida com erro deverá ser retransmitida.

O desafio é selecionar o melhor grupo de nós candidatos para encaminhar as mensagens no menor tempo. A escolha de nós sensores que estejam mais próximos do destino reduz o número de saltos em toda a rota. Por outro lado, nós sensores mais distantes do transmissor receberão os sinais de comunicação com menor intensidade, favorecendo a perda de pacotes.

Neste trabalho é proposto um protocolo cross-layer envolvendo MAC e roteamento que procura identificar os melhores nós candidatos para encaminhar as mensagens em cada salto com a melhor conjugação latência/energia.

1.3 Objetivos

O objetivo deste projeto é a criação de um protocolo *cross-layer anycast* assíncrono, que integra funcionalidades da camada MAC e de roteamento, que permita entregar dados com menor latência e menor consumo de energia. A escolha dos nós do próximo salto é realizada considerando a qualidade do canal e o tamanho do pacote de dado transmitido.

Este protocolo, nomeado de MAC-AHEAD, permite entregar pacotes de dados em regiões que apresentem buracos. O protocolo será baseado no PAX-MAC, contendo duas fases para a entrega de dados: a fase para encontrar a rota e sincronizar os nós e a fase para a entrega dos dados através da rota definida. O protocolo MAC-AHEAD apresenta os seguintes diferenciais em relação ao PAX-MAC:

- Um mecanismo que detecta e retransmite os dados perdidos;
- A capacidade do protocolo transmitir diferentes tamanhos de pacotes selecionando nós do FCS que estejam a uma melhor distância e cuja qualidade do canal de comunicação seja a que permita entregar dados com menor latência e com menor consumo energético;
- Um algoritmo de roteamento que permita ao protocolo contornar regiões com buracos;
- A utilização de um número de nós no FCS variável;

Em suma, o protocolo MAC-AHEAD visa ser um protocolo MAC aplicável e adaptável a terrenos com condições mais próximas dos cenários reais.

1.4 Solução Proposta

O protocolo MAC-AHEAD foi desenvolvido baseado inicialmente no protocolo PAX-MAC.

Um mecanismo que detecta perda de pacotes de dado foi criado no protocolo MAC-AHEAD. Esta característica permitiu ao nó emissor detectar a perda de pacote após decorrido um período de tempo da transmissão ao nó receptor.

Um mecanismo de retransmissão dos dados perdidos foi desenvolvido. Esta característica permitiu que o nó emissor seja preparado novamente para enviar o dado e permitiu também que o nó receptor seja sincronizado novamente para recebê-lo.

Para que o protocolo MAC-AHEAD escolha os nós do FCS que apresentem distâncias e qualidade do canal de comunicação desejada foi realizada uma verificação por parte do nó emissor. Foram escolhidos para compor o FCS os nós vizinhos que estão a uma distância desejada e cuja qualidade do canal de comunicação permita entregar o dado com menor latência e menor consumo de energia.

O protocolo MAC-AHEAD teve a capacidade de contornar buracos a partir da implementação de uma adaptação do protocolo GPSR no modo perímetro. A adaptação foi necessária pois o MAC-AHEAD possui transmissão assíncrona e disseminação de dados *anycast*.

Para permitir a variação do número de nós no FCS foi utilizada uma heurística que relaciona o valor do avanço médio em cada *hop* e o avanço médio teórico.

1.5 Estrutura do Trabalho

O capítulo 2 apresenta o referencial teórico contendo os conceitos utilizados para o desenvolvimento do trabalho. Neste capítulo encontram-se informações gerais relacionados a RSSFs, camada de Controle de Acesso ao Meio (MAC), camada de Rede, design Cross-Layer, protocolo GPSR e protocolo PAX-MAC bem como sobre o modelo de referência IEEE 802.15.4.

O capítulo 3 apresenta alguns trabalhos relacionados. Em grande parte, apresenta os protocolos desenvolvidos que contribuíram para o desenvolvimento do protocolo utilizado como referência deste projeto.

A metodologia do trabalho é apresentado no capítulo 4. Neste capítulo são apresentadas questões relacionados ao desenvolvimento de cada uma das implementações previstas nos obje-

tivos, tais como: desenvolvimento do mecanismo de detecção de erros, desenvolvimento de um mecanismo de retransmissão, integração do protocolo GPSR no modo perímetro ao protocolo MAC-AHEAD e desenvolvimento de uma heurística para a escolha do tamanho de nós no FCS.

O capítulo 5 apresenta sobre as configurações dos testes bem como uma descrição dos testes realizados para a análise de performance do protocolo.

O capítulo 6 apresenta os resultados obtidos e discussões a cerca de cada um dos testes realizados.

A conclusão e propostas de trabalhos futuros são apresentados no capítulo 7.

2 REVISÃO TEÓRICA

Este capítulo apresenta os fundamentos necessários para a compreensão do trabalho proposto.

2.1 Redes de sensores sem fio

Redes de Sensores sem Fio (RSSF) podem ser definidas como sendo sistemas distribuídos de aquisição de dados, utilizados para o monitoramento e vigilância de ambientes. Nas RSSFs os principais componentes de comunicação são os nós sensores que atuam como sendo agentes de sensoriamento e de processamento de dados.

Por meio dos nós sensores, os dados originados do monitoramento vão sendo transmitidos até chegarem a uma estação de base coletora, o *sink*, ou até usuários finais, podendo assim serem analisados. Os nós sensores são constituídos basicamente por uma unidade de sensoriamento e recursos de processamento, armazenamento de informações, fonte de energia e interface de comunicação.

A distribuição dos nós sensores pela rede é feita de maneira a não se pré definir suas localizações. Dessa forma, os protocolos são os responsáveis por realizar a auto-configuração dos nós sensores a partir das diferentes localizações em que foram empregados.

Outra característica importante de uma RSSF é sua capacidade de apresentar esforço cooperativo entre os nós para a entrega de dados. Por apresentarem um processador integrado, cada um dos nós pode realizar cálculos simples localmente a partir de dados a serem enviados e transmitirem apenas os dados necessários e parcialmente processados (AHMED; PENG; WANG, 2007).

As RSSFs são caracterizadas por apresentarem recursos limitados de energia. Dessa forma, a conservação de energia é um dos aspectos mais importantes a serem considerados no projeto e desenvolvimento de RSSF.

2.1.1 Arquitetura de comunicação de RSSF

A pilha de protocolos utilizada em uma RSSF é constituída por camada de aplicação, camada de transporte, camada de rede, camada de enlace, camada física, plano de gerenciamento de energia, plano de gerenciamento de mobilidade e plano de gerenciamento de atividades.

Esta pilha de protocolos apresenta uma combinação entre recursos de energia e roteamento, realiza a integração entre dados e protocolos de rede, realiza a comunicação através do meio sem fio de forma eficiente energeticamente e promove esforços cooperativos dos nós sensores (AKYILDIZ et al., 2002).

Abaixo são apresentadas brevemente as funcionalidades de cada uma das camadas que compõem a pilha de protocolos de RSSF:

- A camada de aplicação é responsável por manter as aplicações em softwares de RSSF. Estas aplicações dependem em grande parte da tarefa de sensoriamento utilizada. Como exemplo de aplicações em RSSF pode-se citar um protocolo de gerenciamento de sensores (SMP), um protocolo de anúncio de dados e atribuição de tarefas (TADAP) e um protocolo de disseminação de dados e consulta de sensores (AKYILDIZ et al., 2002).
- A camada de transporte é responsável por ajudar a manter o fluxo de dados de RSSFs.
- A camada de rede é responsável por rotear os dados provenientes da camada de transporte.
- A camada de enlace de dados foi projetada para lidar com o controle de erros e com o controle de acesso ao meio. Essa camada garante conexões de ponta a ponta confiáveis em uma rede de comunicação (KOCAKULAK; BUTUN, 2017).
- A camada física é responsável pela conversão de fluxos de bits em sinais que são mais adequados para a comunicação através do canal sem fios.
- O Plano de Gerenciamento de Energia gerencia como o nó sensor utiliza energia. Com isto, é possível que um nó sensor não participe do roteamento das mensagens, caso ele esteja com baixo nível de energia.
- O Plano de Gerenciamento de Mobilidade detecta e registra os movimentos dos nós sensores.
- O Plano de Gerenciamento de Tarefas organiza os eventos que estão principalmente monitorando e detectando eventos de uma área específica. Assim, nem todos os nós sensores executam as tarefas de detecção ao mesmo tempo e na mesma área (KOCAKULAK; BUTUN, 2017).

2.1.2 Projeto de RSSF: fatores

O projeto de uma RSSF é influenciado por muitos fatores. A compreensão destes fatores servem como um guia para o desenvolvimento de algoritmos e protocolos para a RSSF bem como para realizar a comparação de diferentes projetos (AKYILDIZ et al., 2002). Os fatores são listados abaixo:

- **Tolerância a Falhas:** é a capacidade de manter as funcionalidades de rede de sensores sem interrupção devido a falha dos nós sensores. Em RSSF, os nós sensores podem apresentar falhas ocasionadas por danos físicos ou por interferência ambiental. Segundo Hoblos, Staroswiecki e Aitouche (2000), a tolerância a falha $R_k(t)$ dos nós sensores é modelada usando uma distribuição de *Poison* para capturar a probabilidade de não ter uma falha no intervalo de tempo $(0,t)$:

$$R_k(t) = \exp(-\lambda_k t) \quad (2.1)$$

onde λ_k e t são respectivamente a taxa de falha de um nó sensor k e o período de tempo. Sendo assim, os algoritmos devem ser projetados para tratar o nível de tolerância a falhas requerido para RSSF, de maneira que as falhas de nós sensores não afetem sua tarefa global.

- **Escalabilidade:** O número de nós sensores utilizados em uma RSSF depende exclusivamente da aplicação, podendo esta possuir milhares de nós sensores em uma rede. Dessa forma, os novos projetos devem ser capazes de trabalhar com um grande número de nós bem como com alta densidade de redes sensores. De acordo com Bulusu et al. (2001) a densidade pode ser calculada pela Equação 2.2.

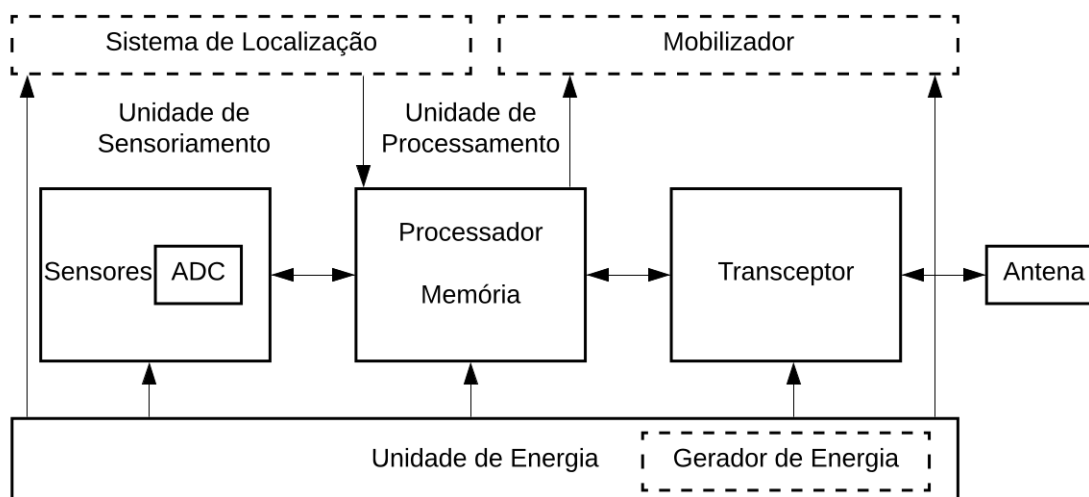
$$\mu(R) = (N\pi R^2)/A \quad (2.2)$$

onde N é o número de nós sensores espalhados na região A ; e R é o alcance de transmissão do rádio. De maneira geral, o $\mu(R)$ fornece o número de nós dentro do raio de transmissão de cada nó na região A .

- **Limitações de Hardware:** Os nós sensores são formados por quatro componentes básicos, sendo eles uma unidade de sensoriamento, uma unidade de processamento, uma

unidade de transmissão e uma unidade de energia, conforme pode ser visto pela Figura 2.1. Dependendo da aplicação, os nós sensores podem ter componentes adicionais, tais como um sistema de posicionamento, um gerador de energia e um módulo de movimento.

Figura 2.1 – Componentes de Nós Sensores



Fonte: Adaptado de (VURAN, 2010)

Algumas aplicações necessitam que estas sub-unidades estejam ajustadas dentro de um único módulo, sendo muitas vezes menores do que um centímetro cúbico. Além do tamanho, existem outras restrições rigorosas para nós sensores, pois eles devem: consumir baixa energia, operar em altas densidades volumétricas, ter baixo custo de produção, ser dispensável, ser autônomo e ser adaptável ao ambiente (AKYILDIZ et al., 2002).

- **Topologia de Redes Sensores:** O termo topologia diz respeito as localizações de cada um dos nós sensores, que de forma conjunta, vão contribuir para a entrega do dado de um nó fonte para um nó destino. Conforme Kocakulak e Butun (2017), a latência, a capacidade e a robustez de uma rede são afetadas por sua topologia. Além disso, a complexidade do roteamento de dados depende da topologia da rede.
- **Ambiente:** Os nós sensores utilizados em uma RSSF são submetidos a diferentes regiões de trabalho, podendo ser utilizados em ambientes com diferentes pressões e temperaturas. Além disso, podem ser empregados em ambientes agressivos como em um campo de guerra ou em ambientes que apresentam interferência.
- **Consumo Energético:** O tempo de vida dos nós sensores apresenta uma forte dependência com o tempo de vida da bateria por ele utilizada. Por esta razão, muitos pesquisadores

estão se concentrando no projeto de protocolos e algoritmos para reduzir o consumo de energia em nós sensores.

Uma vez compreendidos estes fatores, é de extrema importância entender as principais camadas que são objetos para desenvolvimento de protocolos em RSSF, bem como os seus protocolos. Estas camadas são: camada MAC, apresentada na seção 2.2 e camada de roteamento, apresentada na seção 2.4.

2.2 Camada MAC em RSSF

A camada MAC é responsável por realizar o controle de acesso ao canal de comunicação, coordenando a transmissão de canal compartilhado de maneira a definir como e quando um nó realiza a transmissão.

Como descrito, em RSSFs, o consumo de energia é uma grande limitação. Sendo assim, a camada MAC atua com extrema importância, controlando diretamente a operação do transceptor - componente que mais consome energia em um nó sensor. As diferentes fontes de consumo de energia podem ser classificadas nos seguintes (VERMA et al., 2015):

- **Escuta ociosa (*Idle Listening*):** Ocorre quando um nó escuta o canal para uma possível recepção, mas nada é recebido. Este fato é recorrente em redes com baixa carga de tráfego como em RSSFs.
- **Colisões:** Ocorrem quando dois ou mais nós transmitem simultaneamente e o receptor não é capaz de decodificar os pacotes transmitidos. Este problema gera gasto de energia tanto para o remetente, que deve retransmitir os pacotes, quanto para o receptor, que recebe dados que serão descartados.
- **Overhearing:** Ocorre quando um nó sensor desperdiça energia recebendo um pacote que se destina a um destino diferente.
- **Overhead:** Pacotes de dados em RSSFs são geralmente pequenos e portanto, cabeçalhos e mensagens de controle implicam em um alto nível de desperdício de energia para RSSFs.

Os protocolos MAC projetados para redes sem fios locais (WLANs) ou para redes definidas pelo padrão IEEE 802.11 não consideram as limitações recorrentes em RSSF. Dessa

forma, não são adequadas aos requisitos deste tipo de rede. Alguns protocolos MAC tradicionais assumem a existência de uma precisa sincronização de hardware, de recursos ilimitados de energia e realizam escutas continuamente no canal esperando uma possível recepção. Além disso, estes protocolos MAC são focados na taxa de transmissão e latência. Por outro lado, os projetos de RSSFs consomem grande parte dos recursos de energia de um nó sensor devido a escuta ociosa, além do que, em RSSFs a métrica de consumo de energia é mais relevante do que taxa de transmissão.

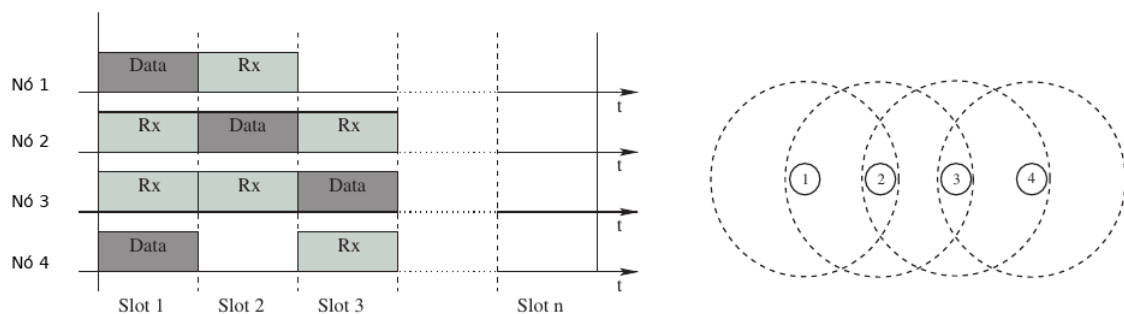
Os tipos de protocolos MAC existentes, são apresentados na seção seguinte.

2.3 Tipos de protocolos MAC para RSSF

Um grande número de protocolos MAC têm sido projetados com o objetivo de reduzir o consumo de energia e evitar assim seu gasto com a escuta ociosa. A abordagem mais utilizada consiste em colocar o transceptor em modo de suspensão durante o máximo de tempo possível, a fim de consumir menos energia. Embora esta abordagem reduza uma grande quantidade de consumo de energia, ela necessita de mecanismos que façam com que um nó sensor desperte caso um nó envie algo para ele. Existem três categorias gerais para abordar esta questão (CANO et al., 2011):

- **Protocolos Programados (*Time Division Multiple Access - TDMA*):** nesta categoria, os segmentos (*slots*) de tempo são atribuídos aos nós sensores dentro de quadros (*frames*) para transmitir ou receber. Dessa forma, os nós contidos nestes segmentos acordam enquanto os outros nós dormem. Nessa abordagem é importante que o nó saiba o tempo de segmento que o nó vizinho vai despertar antes de enviar dados para ele. A representação do funcionamento dos protocolos desta categoria é mostrada na Figura 2.2.

Figura 2.2 – Abordagem TDMA como protocolo MAC

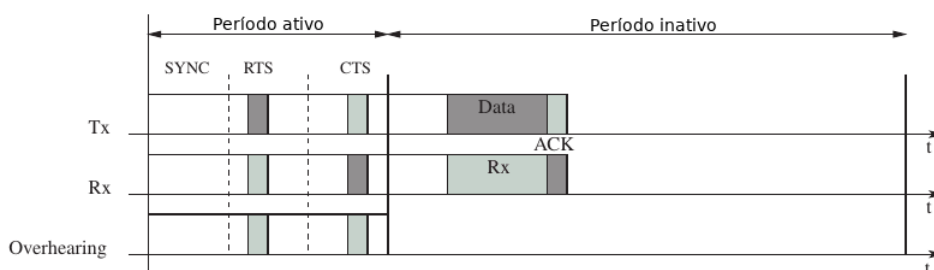


Fonte: Adaptado de (CANO et al., 2011)

Pode-se perceber que os nós transmitem em seus segmentos e acordam para receber nos segmentos de seus vizinhos. Protocolos programados apresentam maior controle de atrasos, maior taxa de transferência e reduzem ou anulam as probabilidades de colisão, exigindo para isto, grande sincronização dos nós sensores. Nesta categoria, em Kaur e Kumar (2016) são descritos e analisados alguns protocolos, sendo alguns deles: E-TDMA (HEINZELMAN; CHANDRAKASAN; BALAKRISHNAN, 2002), BMA (LI; LAZAROU, 2004) e BSMAC (ALVI et al., 2015).

- **Protocolos com Períodos Ativos Comuns:** estes protocolos organizam os nós sensores para dormirem e acordarem no mesmo instante de tempo. Desta forma, o nó sensor deve esperar para o próximo período acordado do nó para enviar o dado. Um dos protocolos mais conhecidos desta categoria é o protocolo S-MAC (YE; HEIDEMANN; ESTRIN, 2004). A representação de sua funcionalidade pode ser vista na Figura 2.3:

Figura 2.3 – Protocolo com períodos Ativos Comuns



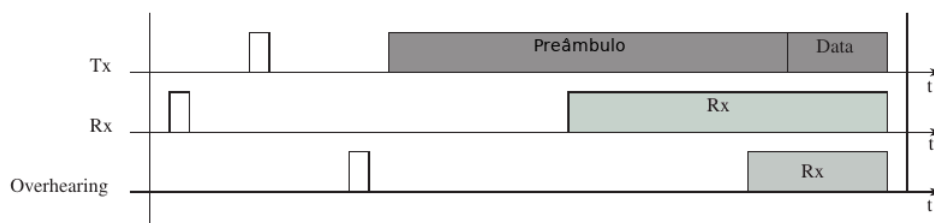
Fonte: Adaptado de (CANO et al., 2011)

Observa-se que os nós sensores prontos para transmitirem esperam até que o período esteja ativo. No protocolo S-MAC, o primeiro *handshake* composto por RTS/CTS é executado no período ativo, mas a comunicação dos dados é feita no período de suspensão. Os protocolos com períodos ativos comuns apresentam custo associado com a criação e manutenção da programação quando o nó dorme e acorda.

- **Protocolos MAC Assíncronos:** Nesta categoria os nós sensores acordam e dormem independentemente dos outros, não sendo necessário nenhuma sincronização. Mecanismos para se certificar de que os receptores estarão acordados para receberem dados são implementados. Os protocolos dividem-se em amostragem de preâmbulos e abordagens iniciadas pelo receptor. Na amostragem de preâmbulo, os nós sensores acordam e permanecem acordados apenas para verificar se o canal está ocupado. Em seguida, um nó sensor que

deseja transmitir envia um preâmbulo longo para sobrepor-se ao tempo de amostragem de canal do receptor e em seguida envia os dados. O comportamento da amostragem por preâmbulos pode ser representado pela Figura 2.4

Figura 2.4 – Protocolo MAC Assíncrono (Amostragem de Preâmbulos)



Fonte: Adaptado de (CANO et al., 2011)

Pode ser observado que o preâmbulo longo faz com que o receptor e um nó de escuta permaneçam acordados para receber a mensagem. Por outro lado, em protocolos iniciados pelo receptor, os nós sensores enviam um anúncio quando acordam. Em seguida, um nó sensor com um pacote transmite e espera até receber o anúncio enviado pelo receptor. Em Bachir et al. (2010) são descritos alguns protocolos nesta categoria, dentre tantos, tem-se: B-MAC (POLASTRE; HILL; CULLER, 2004), X-MAC (BUETTNER et al., 2006) e GeRAF (ZORZI; RAO, 2003).

O protocolo MAC-AHEAD é um protocolo assíncrono baseado no protocolo PAX-MAC. O protocolo PAX-MAC é apresentado na seção 3.2.

2.4 Camada de rede em RSSF

Conforme apresentado anteriormente, uma grande limitação das RSSFs é o consumo energético. Desta forma, é altamente necessário o desenvolvimento de projetos que combinem a utilização de um grande número de nós sensores na rede e técnicas que reduzam o desperdício de energia responsável pelo encurtamento da vida de uma RSSF.

Na camada de rede é importante a utilização de métodos que encontrem a rota com eficiência energética e que realize retransmissões de dados dos nós sensores para a estação base de maneira que a vida útil da rede seja maximizada (AL-KARAKI; KAMAL, 2004). O desenvolvimento de protocolos para a camada de rede em RSSF é bastante desafiador devido as suas diferentes características em relação a outras redes sem fio. Segue algumas destas características que as diferenciam:

- Grande parte das aplicações de redes sensores requerem um fluxo de dados detectados a partir de fontes múltiplas para uma estação de base particular, ao contrário de outras redes de comunicação típicas. No entanto, não impede que o fluxo de dados esteja em outras formas, como por exemplo em *multicast* ou *peer to peer*.
- RSSF requerem um cuidadoso gerenciamento de recurso por apresentarem limitação de energia.
- Na maioria dos cenários de aplicação, os nós em RSSFs são estacionários após a implantação. Em redes sem fios tradicionais os nós são livres para se mover, resultando em mudanças topológicas imprevisíveis e frequentes.

Por estas diferenças, novos algoritmos foram propostos para o problema de roteamento em RSSFs. Esses mecanismos de roteamento levaram em consideração os recursos inerentes à este tipo de rede, juntamente com os requisitos de arquitetura e aplicativos. A tarefa de encontrar e manter rotas em RSSFs não é trivial, pois as restrições de energia e as mudanças repentinas no estado do nó provocam alterações topológicas frequentes (AL-KARAKI; KAMAL, 2004).

O tipos de protocolos de roteamento utilizados em RSSF são apresentados na seção seguinte.

2.4.1 Tipos de protocolos da camada de rede em RSSF

Para minimizar o consumo de energia, as técnicas de roteamento utilizadas em RSSFs presentes na literatura utilizam táticas já conhecidas de roteamento bem como táticas de agregação de dados, processamento em rede, clusterização e atribuição de métodos centrado em dados.

Uma taxonomia apresentada por Patil e Biradar (2012) para os protocolos de roteamento, segundo vários critérios de classificação são: centrado em dados, hierárquicos, baseado em localização, baseado em negociação, baseado em múltiplos caminhos, baseado em qualidade de serviço e baseado em mobilidade.

No roteamento centrado em dados, os protocolos são capazes de selecionar um conjunto de nós sensores e utilizar a agregação de dados durante o processo de transmissão. Alguns atributos são utilizados para especificar as propriedades dos dados e reduzir a redundância durante o processo de transmissão.

No roteamento hierárquico, os nós são organizados em uma arquitetura hierárquica dentro de clusters. Os nós com baixa energia são utilizadas para realizar o sensoriamento e enviar dados para o *cluster head* a curtas distâncias. Os nós que apresentam uma quantidade maior de energia são utilizados como *clusters head* para agregar os dados de seus membros e enviar para o *sink*. Este processo reduz o consumo de energia, realiza balanceamento de carga e melhora a escalabilidade. Alguns dos protocolos desta categoria são: LEACH (HANDY; HAASE; TIMMERMANN, 2002), TEEN (MANJESHWAR; AGRAWAL, 2001), SEP (SMARAGDAKIS; MATTA; BESTAVROS, 2004).

No roteamento baseado em localização, a posição geográfica dos nós é importante pois o roteamento é feito baseado na localização, endereçando cada um dos nós. Alguns dos protocolos baseados em localização são: GAF (XU; HEIDEMANN; ESTRIN, 2001), GPSR (KARP; KUNG, 2000) e GeRaF (ZORZI; RAO, 2003).

No roteamento baseado em negociação, é utilizado um grande número de descritores de dados chamados de metadados a fim de evitar a transmissão de dados redundantes através da negociação. As decisões de comunicação também são tomadas com base nos recursos disponíveis. Um exemplo de protocolo baseado em negociação é o CSPIN (PHAM et al., 2010).

Os protocolos de roteamento baseado em múltiplos caminhos utilizam diferentes caminhos para melhorar o desempenho da rede e a tolerância as falhas. Estes caminhos são mantidos operantes enviando mensagens periódicas (GANESAN et al., 2001).

Os protocolos baseados em qualidade de serviço surgem também com a proposta de minimizar o consumo de energia considerando os requisitos em termos de qualidade de serviço, confiabilidade e tolerância as falhas. Alguns dos protocolos que introduz a noção de qualidade de serviço são apresentados em Sumathi e Srinivas (2012).

Os protocolos de roteamento que apresentam mobilidade são utilizados em RSSFs que necessitam que os nós sensores estejam em movimento acompanhado da tarefa de sensoriamento. Alguns protocolos de roteamento que utilizam *sinks* móveis são apresentados em (GOYAL; KHANDNOR, 2017).

Uma ampla gama de protocolos de roteamento vem sendo utilizados em *Internet of Things* (Beevi, 2016), sendo o RPL (THUBERT et al., 2012) um deles.

O protocolo MAC-AHEAD utiliza um roteamento baseado em localização, implementando uma adaptação do protocolo GPSR, apresentado na seção 3.3. A forma de disseminar

mensagens utilizada pelo MAC-AHEAD é o *anycast*. Os métodos de disseminação de mensagens são apresentados na subseção 2.4.2 a seguir.

2.4.2 Disseminação de Mensagens

As diferentes formas de disseminação de mensagens definem a maneira como os dados e mensagens de controle são entregues a cada um dos *hosts* ou nós sensores de uma RSSF.

Na disseminação *Unicast* ocorre o envio de mensagens entre um nó fonte e um único nó destino.

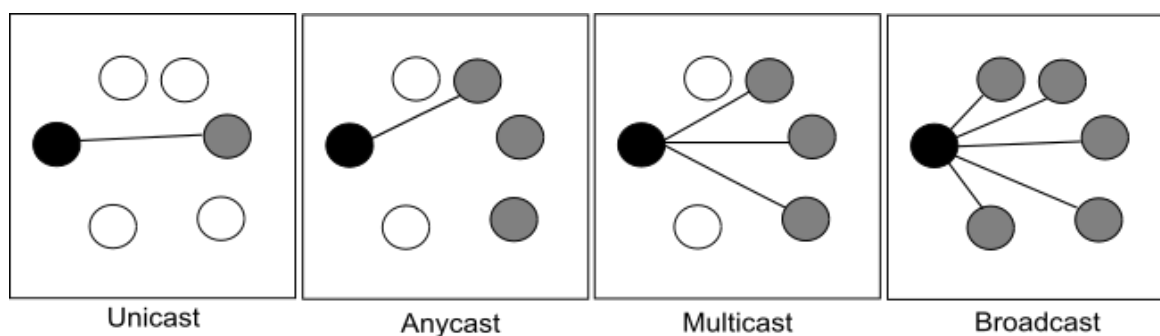
Na disseminação *Anycast* ocorre o envio da mensagem para qualquer nó pertencente a um grupo de nós. Esse nó é escolhido utilizando algum parâmetro específico.

Na disseminação *Multicast*, as mensagens são enviadas para todos os membros de um determinado grupo. Por exemplo, um grupo de processos que implementa um sistema de bancos de dados distribuídos.

Na disseminação *Broadcast* os *hosts* precisam enviar mensagens a todos os outros *hosts*. Por exemplo, atualizações do mercado de ações pode funcionar melhor pela difusão das informações a todas as máquinas, permitindo que aquelas que estivessem interessadas lessem os dados.

A Figura 2.5 exibe cada uma destas formas de se disseminar mensagens.

Figura 2.5 – Disseminação de mensagens. Nós com cores pretas são os nós transmissores da mensagem.



Fonte: do autor (2019)

Uma das estratégias utilizadas para reduzir o consumo de energia e acelerar a entrega do dado em RSSF é utilizar a estratégia de *duty-cycle* com disseminação *anycast*. Com *duty-cycle* os nós sensores alternam entre ciclos de ativo/inativo ao longo da transmissão e a disseminação *anycast* reduz o tempo de espera de um nó emissor encontrar o nó do próximo *hop*.

Outros tipos de protocolos utilizados em RSSF são protocolos que utilizam um projeto *cross-layer*. A seguir é apresentado sobre esta abordagem.

2.5 Projeto *cross-layer*

O modelo OSI (*Open Systems Interconnect*) define uma hierarquia de serviços individuais a serem implementados pelos protocolos para cada uma das cinco camadas.

Na fase de desenvolvimento de um protocolo, o projetista pode ter duas escolhas. Na primeira escolha, os protocolos podem ser projetados respeitando as regras da arquitetura original. No caso do modelo de referência OSI em camadas, isso significaria projetar protocolos de modo que eles usem apenas os serviços nas camadas inferiores e não se preocupem com os detalhes de como o serviço está sendo fornecido. Na segunda escolha, os protocolos podem ser projetados violando a arquitetura de referência. Como as arquiteturas de referência em comunicação e rede tradicionalmente foram colocadas em camadas, esta violação é geralmente denominada como projeto *cross-layer*.

Segundo Srivastava e Motani (2005), projeto *cross-layer* se refere a um projeto de protocolo que viola a arquitetura de comunicação em camadas. O termo violar significa não projetar um protocolo nas diferentes camadas independentemente impondo algumas condições de processamento nas outras camadas.

Dado um modelo hipotético de três camadas com as camadas denotadas por L1, L2 e L3, sendo L1 a camada mais baixa e L3 a mais alta. Pode-se projetar um protocolo L3 que precise de L1 para passar um parâmetro para L3 em tempo de execução. Isso exige uma nova interface, não disponível na arquitetura. Alternativamente, pode-se considerar L2 e L1 como uma única camada e projetar um protocolo conjunto para essa “super camada”. Ou, pode-se projetar o protocolo em L3, tendo em mente o processamento que está sendo feito na L1, novamente dando o luxo de projetar os protocolos nas diferentes camadas independentemente. Todos esses são exemplos de projeto *cross-layer* em relação à arquitetura original (SRIVASTAVA; MOTANI, 2005).

2.6 Controle de erros em RSSF

O controle de erros é necessário em qualquer comunicação de dados, para que as mensagens cheguem corretamente ao destino. Nas RSSFs, é preciso ter formas de evitar erros de transmissão, pois retransmissões aumentam a latência e o consumo de energia.

De maneira geral, os mecanismos de controle de erros podem ser categorizados em três abordagens principais, sendo elas:

- **Automatic Repeat Request (ARQ):** ou Pedido de Requisição Automática, o controle de erros depende principalmente da retransmissão para a recuperação dos pacotes de dados perdidos. Os protocolos ARQ permitem a retransmissão de pacotes com falhas enviando uma confirmação explícita após a falha. Possui a desvantagem de gerar custos de retransmissões adicionais significativos e despesas gerais em caso de erros. Por outro lado, em caso de boa qualidade do canal, a sobrecarga dos protocolos ARQ é baixa comparada aos esquemas de corretores de erro progressivo *Forward Error Correction (FEC)*. A eficiência do ARQ em aplicações de rede sensores (RS) é limitada devido à escassez de energia e recursos de processamento dos nós de sensores (VURAN; AKYILDIZ, 2009).
- **Forward Error Correction (FEC):** ou Corretores de Erro Progressivo, adiciona redundância ao pacote transmitido de maneira que ele possa ser recebido no receptor sem erros. Apresentam a desvantagem de gerar sobrecarga de comunicação em termos de transmissão e recepção de bits redundantes adicionais, bem como pacotes de decodificação. Em RSSF, a energia de decodificação também deve ser considerada na avaliação do desempenho dos protocolos FEC devido as baixas taxas de clock usadas pelos CPUs.
- **ARQ Híbrido (HARQ):** este mecanismo visa explorar as vantagens dos esquemas FEC e ARQ aumentando de maneira incremental a resiliência de erro de um pacote através de retransmissões. Existem dois tipos de esquemas HARQ: Tipo I e Tipo II. Utilizando HARQ-I, primeiro é enviado um pacote não codificado ou um pacote codificado com uma menor capacidade de correção de erros. Caso este pacote seja recebido por engano, o receptor envia uma confirmação negativa (NACK) para o remetente, que reenvia o pacote codificado com um código FEC mais poderoso. A diferença no Tipo II é que para retransmissões, apenas os bits redundantes são enviados. Enquanto o Tipo II diminui o uso de largura de banda do protocolo, o Tipo I não requer que os pacotes enviados anteriormente sejam armazenados (VURAN; AKYILDIZ, 2009).

Utilizando a abordagem (FEC) e esquemas ARQ híbridos melhoram a resiliência de erro em comparação com esquemas ARQ pois estes enviam bits redundantes através do canal sem fio. Portanto, podem ser suportados valores de relação sinal/ruído (SNR) menores para obter a mesma taxa de erro que uma transmissão não codificada.

O protocolo MAC-AHEAD utiliza uma abordagem ARQ, uma vez que realiza a retransmissão para recuperação dos pacotes perdidos. Esta escolha é devido ao fato do protocolo ter

um mecanismo que permite escolher os nós baseado na qualidade do canal de comunicação, o que reduz o número de retransmissões.

2.7 Modelo de referência para RSSFs IEEE 802.15.4

O padrão para redes sem fio com transmissões utilizando baixa frequência é definido pelo modelo de referência IEEE 802.15.4. No seu escopo, estão definidas as especificações da camada física e da camada de Controle de Acesso ao Meio (MAC) para conectividade sem fio com baixa taxa de dados com dispositivos físicos, portáteis, que não possuem baterias ou que apresentam recursos de energia limitados. As especificações da camada física são definidas para dispositivos de baixa potência (mW) que operam em faixas de frequências não licenciadas, denominadas ISM (*Industrial, Scientific and Medical*). A taxa transmissão varia de 20kbps a 250kbps.

O propósito do padrão é fornecer baixa complexidade, baixo custo, baixo consumo de energia e acesso a redes sem fio com baixa taxa de dados entre dispositivos baratos. Define também as diferentes bandas de frequência suportadas pelas camadas físicas (IEEE, 2003).

O simulador utilizado nos experimentos é configurado para ter um modelo de rádio baseado no modelo de referência IEEE 802.15.4.

3 TRABALHOS RELACIONADOS

A seguir são apresentados os trabalhos relacionados ao projeto.

3.1 Protocolos

Muitos protocolos tem sido desenvolvidos para RSSF visando reduzir o consumo de energia e a latência. Dois protocolos MAC pioneiros para RSSF são o S-MAC e o B-MAC descrito respectivamente em Ye, Heidemann e Estrin (2002) e Polastre, Hill e Culler (2004). Estes protocolos utilizam o método duty-cycling como principal forma de economizar energia. O método duty-cycling mantém o nó em modo de suspensão a maior parte do tempo e o torna ativo para executar aquisição de dados, computação e comunicação em curtos espaços de tempo.

No S-MAC, os nós acordam em períodos ativos comum, trocam os dados e voltam ao modo de suspensão. Desta forma, ocorre o consumo de energia mesmo quando não há dados a serem transmitidos. O B-MAC é um protocolo MAC assíncrono, que utiliza a transmissão de longos preâmbulos para estabelecer a comunicação. Nesta abordagem, quando o receptor detecta atividade no canal, ele mantém o rádio ligado até receber o dado depois do preâmbulo. Uma grande desvantagem do B-MAC é que muita energia é gasta, pois nós que não fazem parte da transmissão ficam acordado até o final do preâmbulo.

Muitas abordagens de protocolos MAC duty-cycling foram propostas com o objetivo de alcançar atrasos mínimos e alta eficiência energética. Lu, Krishnamachari e Raghavendra (2004) apresenta o D-MAC, um protocolo baseado no S-MAC que não utiliza períodos ativos comuns. Nesta abordagem, o agendamento é escalonado na direção da estação base, de forma que os nós mais próximos a ela são acordados mais tarde dos que os nós que estão mais distante.

Outra abordagem baseada no S-MAC é apresentado em Du, Saha e Johnson (2007). O R-MAC envia mensagens de dados separadas das mensagens de controle, de maneira que quando os nós estão acordados, o canal de comunicação é reservado para a transmissão de dados durante o período inativo comum. No tempo de suspensão, apenas os nós que participarão na comunicação despertarão no horário programado para receber e enviar dados. Os resultados da simulação apresentaram uma redução de 6 vezes na latência do RMAC em relação ao S-MAC.

Conforme apresentado em Buettner et al. (2006), o X-MAC é um protocolo para redes assíncronas que envia uma sequencia de preâmbulos curtos introduzindo intervalos de escuta entre eles, de modo a permitir que o receptor notifique a sua presença. Desta forma, assim que o emissor detecta a notificação do receptor, interrompe a transmissão dos preâmbulos curtos

e transmite o pacote de dados. X-MAC mitiga o problema relacionado ao overhearing do B-MAC e, ao mesmo tempo, reduz pela metade o *sleep-delay* - problema do aumento da latência ocasionado pelo período de inatividade do nó sensor.

Mais recentemente, as abordagens *cross-layer* propostas visam poupar mais energia em comparação com as tradicionais (MENDES; RODRIGUES, 2011). A fim de mitigar o *sleep-delay* em RSSF que utilizam *duty-cycling*, muitos protocolos *cross-layer* combinam funções MAC e funções de camadas de rede, explorando conceitos de roteamento geográfico e combinado com MAC baseado no preâmbulo.

GeRaF, conforme descrito em Zorzi e Rao (2003), foi uma proposta pioneira entre os protocolos *anycast* que visa reduzir a latência em redes assíncronas usando sondagem. A vizinhança do nó remetente é dividida em regiões e, a partir daí, o remetente começa a pesquisar a região mais próxima do destino usando uma mensagem de preâmbulo. O primeiro nó que responde com um sinal eACK é selecionado como o melhor encaminhador. Caso não haja nó acordado nessa região, o remetente procede a procurar a próxima região com um pacote de continuação especial, específico para apenas essa região.

O CMAC, apresentado em (LIU; FAN; SINHA, 2007), realiza uma melhoria do GeRaF, eliminando a transmissão de mensagens de controle intermediárias. CMAC define um avanço mínimo exigido (r_0) como um critério para selecionar os membros do FCS. Os vizinhos do remetente 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 dos dados não é considerado.

No AGA-MAC, apresentado em (HEIMFARTH; GIACOMIN; ARAUJO, 2015), é imposto um limiar no avanço para o destino, como no CMAC, com a diferença de que este limiar depende do comprimento dos dados. Se a carga útil dos dados é curta, o limiar é reduzido para reduzir o tempo de avanço. Se a carga útil dos dados for longa, o limiar é ampliado para reduzir os saltos necessários para alcançar o destino. Os autores demonstraram que existe um valor de limiar ótimo para cada comprimento de pacote de dados que proporciona a menor latência na rede.

3.2 PAX-MAC: Protocolo assíncrono e *anycast* de preâmbulos antecipados para RSSF

O PAX-MAC é um protocolo *cross-layer* para RSSF que integra funções da camada de rede e funções da camada MAC. O principal objetivo é reduzir a alta latência em protocolos

MAC assíncronos de RSSF (HEIMFARTH et al., 2016). A transmissão é assíncrona e utiliza técnicas de despertar e suspender um nó sensor (duty cycling) como principal forma de economizar energia.

Este protocolo transmite os dados separadamente dos preâmbulos. Os preâmbulos são enviados à frente do pacote de dados para criar uma rota até o destino e marcar os horários de transmissão da mensagem de dados até o destino. Para isto, os preâmbulos contêm a informação do tempo em que o dado será transmitido para o próximo nó sensor. Deste modo, os dados são transmitidos através do percurso escalonado pelos preâmbulos. Simultaneamente, os preâmbulos estão programando os nós sensores alguns saltos à frente.

Na transmissão de preâmbulos, o nó remetente primeiramente define o número de nós sensores para o conjunto de nós encaminhadores, *Forwarding Candidate Set* (FCS). Logo em seguida, é definido quais de seus nós vizinhos farão parte do FCS. Para isto, é utilizado o algoritmo guloso de roteamento geográfico na forma *anycast*. A partir daí, o nó emissor inicia a transmissão em *broadcast* de preâmbulos (mensagens RTS) para seus nós vizinhos, sendo que apenas os nós pertencentes ao FCS podem responder. Nestes preâmbulos contêm a informação do tempo que será enviado o dado.

Quando um dos nós pertencentes ao FCS acorda e recebe a mensagem RTS, este armazena a informação de tempo que deverá estar acordado para receber o dado, cria uma mensagem RTS e inicia o envio de preâmbulos a fim de continuar o processo de escalonamento. Quando o remetente é o nó destino, este cria uma mensagem CTS e envia para o nó emissor como uma confirmação da mensagem RTS recebida.

Após um intervalo de tempo τ , o primeiro nó emissor envia o dado para o segundo nó do escalonamento. O segundo nó acorda no tempo que tinha sido armazenado e recebe a mensagem. A partir daí, o dado é transmitido para os nós a frente obedecendo o escalonamento criado durante a emissão de RTS. Caso o nó receptor seja o nó de destino, este nó cria uma mensagem ACK e envia para o emissor como uma confirmação do dado que foi entregue.

A distância de tempo entre o pacote de dados e a série de preâmbulos é coordenada pelo nó de origem.

3.2.1 Seleção do FCS

Os nós vizinhos que fornecem os maiores avanços até o nó destino são escolhidos para compor o FCS, pois com esta abordagem, os números de saltos serão reduzidos e menos energia

será utilizada para a transmissão de dados. Para isto é utilizado o protocolo GPSR (KARP; KUNG, 2000) no modo guloso.

3.2.2 Cardinalidade do FCS

O número de membros FCS determina o número médio de preâmbulos para entrar em contato com o próximo nó transmissor. No PAX-MAC, o tempo para transmitir esses preâmbulos deve ser menor ou igual à duração da transmissão do pacote de dados. A cardinalidade FCS é obtida pela Equação 3.1:

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

onde v é o número de vizinhos, p é igual a média do número de preâmbulos e N_p igual ao número máximo de preâmbulos enviado.

A cardinalidade do FCS é fixa para cada um dos diferentes tamanhos de pacotes transmitidos e utilizada para controlar a velocidade de emissão dos preâmbulos.

Abaixo é apresentado o protocolo de roteamento geográfico GPSR, utilizado como referência para o desenvolvimento do algoritmo que permite contornar regiões com buracos.

3.3 Greedy Perimeter Stateless Routing (GPSR)

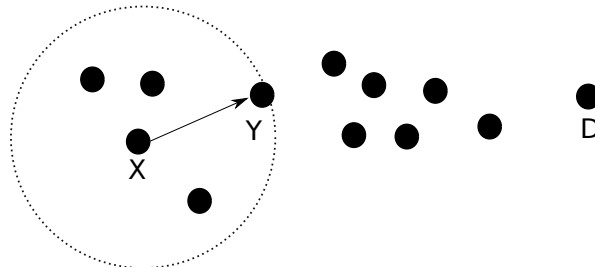
Greedy Perimeter Stateless Routing (GPSR) é um protocolo que utiliza a posição dos nós sensores e o destino do pacote de dados para decidir qual a melhor decisão de encaminhamento. Neste protocolo, o encaminhamento do pacote pode ser realizado no modo guloso (*greedy*) ou no modo perímetro (*perimeter*). O protocolo MAC-AHEAD utiliza um roteamento baseado no GPSR.

3.3.1 Modo guloso (*greedy*)

O protocolo GPSR no modo guloso encaminha o pacote para os nós sensores que apresentam o maior avanço em direção ao nó de destino. Para isto, o protocolo GPSR, a partir de um nó sensor, calcula a distância euclidiana entre cada um dos seus nós vizinhos e o nó de destino, uma vez que a localização destes nós é conhecida. O nó sensor vizinho que apresentar a menor distância euclidiana, portanto o maior avanço até o destino, será escolhido como o nó receptor. A Figura 3.1 apresenta a escolha de encaminhamento realizado pelo nó x quando utilizando o

modo guloso. Neste cenário, o círculo tracejado indica o alcance de comunicação do rádio do nó x. Desta forma o nó x possui 4 nós vizinhos, sendo que o nó y é o mais próximo do nó de destino D.

Figura 3.1 – Exemplo de encaminhamento no modo guloso. O nó x encaminha dado para o nó y.

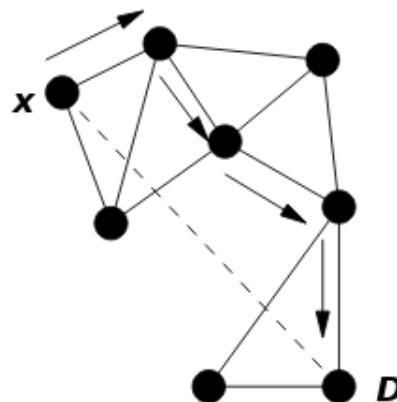


Fonte: Adaptado de (KARP; KUNG, 2000)

3.3.2 Modo perímetro (*perimeter*)

Em cenários que ocorra a falha da execução do GPSR no modo guloso, é utilizado o encaminhamento no modo perímetro. Nesta fase, o encaminhamento é realizado de maneira a contornar o buraco a partir de sua borda. Para que isto seja possível, é utilizado um algoritmo que percorra uma região poligonal fechada no sentido anti-horário (regra da mão direita). A cada aresta escolhida na regra da mão direita, esta não deve intersectar a aresta formada pelo nó que ocorreu o ponto de falha e nó de destino final. A Figura 3.2 apresenta o nó x, um possível ponto de falha, realizando o roteamento no modo perímetro até o nó de destino D. As setas indicam as arestas escolhidas para o roteamento, sendo estas não interceptadas pela aresta xD.

Figura 3.2 – Nó x realizando o roteamento no modo perímetro para o nó de destino final D.



Fonte: Adaptado de (KARP; KUNG, 2000)

3.3.3 Execução do protocolo GPSR

Por ser um protocolo geográfico, cada um dos nós da rede possui informações da localização de seus nós vizinhos e do nó de destino.

Para permitir o contorno de buracos e operar no modo perímetro, o cabeçalho do pacote deste protocolo possui alguns atributos específicos conforme apresentado na Tabela 3.1. Um dos atributos adicionados é utilizado para indicar quando o protocolo está no modo guloso ou no modo perímetro M . Todos os pacotes iniciam com o modo guloso.

Tabela 3.1 – Cabeçalho do pacote GPSR (KARP; KUNG, 2000)

Campo	Função
D	Localização do nó destino
Lp	Localização do pacote que entrou no modo perímetro
Lf	Última aresta de envio do pacote
$e0$	Primeira aresta após ocorrer a falha do modo guloso
M	Modo do pacote: guloso ou perímetro

Uma vez recebido um pacote no modo guloso para encaminhamento, o nó procura por um nó vizinho mais próximo geograficamente do destino. Caso possua um nó vizinho próximo, o nó encaminha o pacote para este nó vizinho.

O GPSR encaminha os pacotes no modo perímetro usando um simples percurso de gráfico planar. Desta forma, supondo que o protocolo tenha entrado no modo perímetro no nó x e que possua como nó destino o nó D , o GPSR encaminha os dados em faces progressivamente mais próximas do gráfico planar, cada uma das quais é atravessada pela linha xD .

Quando o pacote entra no modo perímetro, o GPSR registra a localização que ocorreu a falha do encaminhamento no modo guloso (Lp) no pacote. Esta localização é usada para determinar quando o pacote pode retornar para o modo guloso. Para cada novo encaminhamento realizado, o GPSR registra em Lf os nós formados por nó emissor e nó receptor da nova face. A primeira face utilizada para encaminhamento é registrada em $e0$.

Ao receber um pacote no modo perímetro, o GPSR compara a localização de Lp com a localização do nó atual. Sendo a distância do nó atual até D menor do que a distância de Lp a D o pacote é retornado para o modo guloso.

Ao entrar no modo perímetro em x , o GPSR encaminha o pacote para a primeira aresta no sentido anti-horário de x para D . Logo em seguida, o GPSR encaminha o pacote em torno da face pela regra da mão direita.

Quando x e D são conectados pelo gráfico, para atravessar a face que limita x em qualquer direção utiliza-se a regra da mão direita. Esta deve levar o pacote a um ponto y no qual xD cruza o lado oposto da face. Em y , o GPSR reduziu claramente a distância entre o pacote e o seu destino, em comparação com o início do pacote no modo perimétrico em x .

Durante o encaminhamento em torno de uma face da aresta, o GPSR determina se a borda para o próximo salto escolhido intercepta o xD . O GPSR possui as informações necessárias para fazer essa determinação, uma vez que Lp e D são gravadas no pacote, e um nó GPSR armazena sua própria posição e as de seus vizinhos.

Em cada face, o pacote progride pela regra da direita até atingir a borda que intercepta com xD em um ponto mais próximo do campo Lf do pacote para D . Finalmente, a face contendo D é alcançado, e a regra da mão direita leva a D ao longo dessa face.

Quando D não é acessível o GPSR encaminhará um pacote de modo perímetro até que o pacote atinja uma face acessível. Ao chegar a esta face interior ou exterior, o pacote irá rodar sem sucesso em torno da totalidade da face, sem encontrar uma borda que se entrem xD em um ponto mais próximo de D do que Lf . Quando o pacote atravessa a primeira aresta que ocupou essa face pela segunda vez, o GPSR percebe a repetição do encaminhamento na borda e0 armazenada no pacote e o pacote é descartado corretamente, pois o destino não é alcançável; o cruzamento no modo perímetro de gráfico para um destino alcançável nunca envia um pacote através do mesmo link na mesma direção duas vezes.

Para cada um dos protocolos apresentados, não se identificou a presença de um mecanismo adaptativo aos erros do canal de comunicação.

4 PROTOCOLO MAC-AHEAD

Neste capítulo são apresentadas as características do protocolo MAC-AHEAD.

4.1 Visão geral

O protocolo MAC-AHEAD é um protocolo *cross-layer* assíncrono *anycast* que integra funcionalidades da camada MAC e de roteamento. Seu objetivo é reduzir a latência existente nos protocolos MAC assíncronos devido ao problema do *sleep-delay*. Para economizar energia, os nós operam em ciclos ativo/inativo, sem sincronização. Os protocolos de amostra de preâmbulo utilizam o mecanismo a seguir para permitir a transmissão em uma rede assíncrona: quando o remetente tem uma mensagem, ele começa a enviar uma série de mensagens curtas para encontrar um nó receptor.

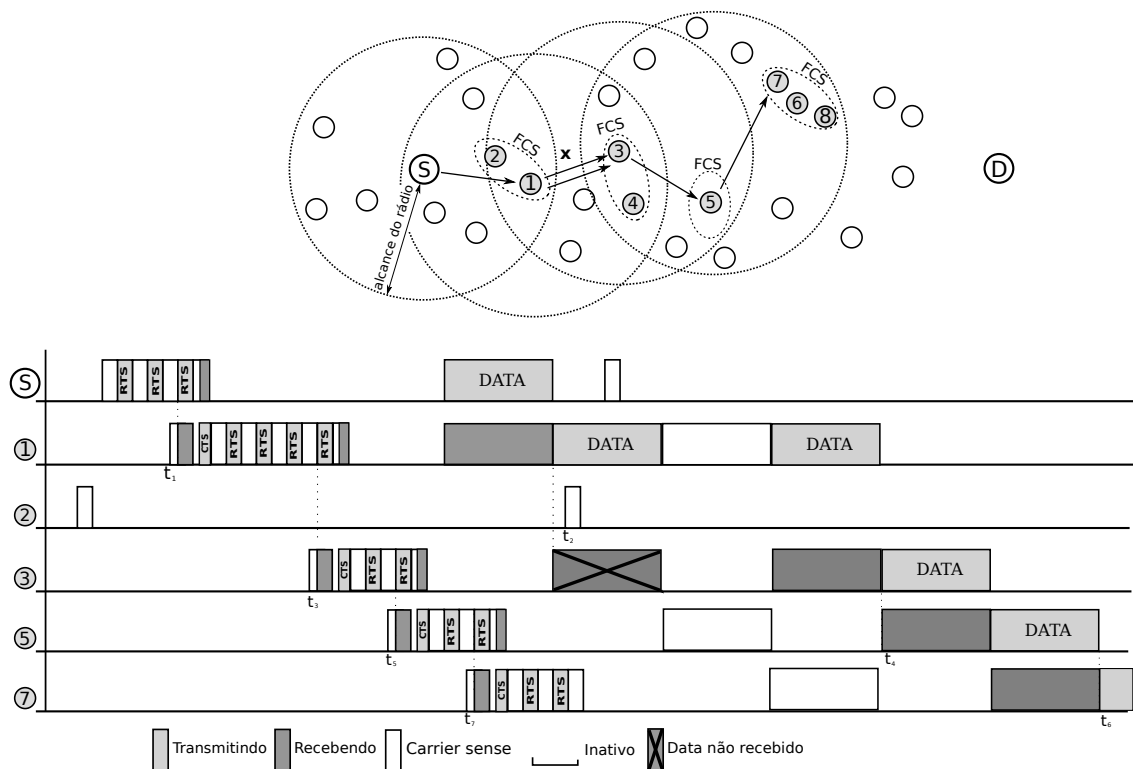
Nosso protocolo opera no modo *anycast*: em vez de encaminhar a mensagem para um único e determinado nó receptor, o protocolo seleciona um conjunto de vizinhos com a capacidade de transmitir os dados (*Forwarding Candidate Set* - FCS). O primeiro nó ativo deste conjunto envia resposta em um $eACK$ para informar sua disponibilidade para atuar como próximo salto. A série de preâmbulos é interrompida e o nó age como próximo nó. O uso do método *anycast* reduz o tempo médio para encontrar o nó do próximo salto, já que apenas um único nó do conjunto deve estar ativo para ser selecionado. Para selecionar um subconjunto de nós adequado como FCS, um algoritmo de roteamento geográfico é empregado: o subconjunto compreende os nós com maior avanço em direção ao destino com uma relação sinal-ruído adequada.

O MAC-AHEAD não envia dados após o sinal $eACK$ como outros protocolos MAC convencionais de amostragem de preâmbulo. Para transmitir os dados, o protocolo utiliza duas fases. Na primeira, os preâmbulos são propagados à frente para descobrir a rota e marcar o tempo para a transmissão de dados. Na segunda fase, o pacote de dados é transportado pela rede pelos nós anteriormente escalonados. Essas duas fases são iniciadas com um deslocamento do tempo. Isto significa que os preâmbulos estão sendo transmitidos antecipadamente para despertar nós e reservar o canal de comunicação para o pacote de dados. Ao mesmo tempo, o pacote de dados está sendo transmitido em uma área espacialmente separada da rede por nós previamente agendados pelos preâmbulos. É necessário cuidado para evitar colisões entre preâmbulos e pacote de dados, uma vez que os preâmbulos não se propagam a velocidade constante.

Um parâmetro muito importante neste esquema é a distância entre o pacote de dados e a série de preâmbulo. Isso é realizado devido a um atraso inicial no nó de origem entre o envio de preâmbulos e dados. Esse tempo deve ser o suficiente para evitar colisões, porém atrasos maiores aumentam a latência geral.

A Figura 4.1 descreve o funcionamento do protocolo MAC-AHEAD.

Figura 4.1 – Visão geral do protocolo MAC-AHEAD.



Fonte: do autor (2019)

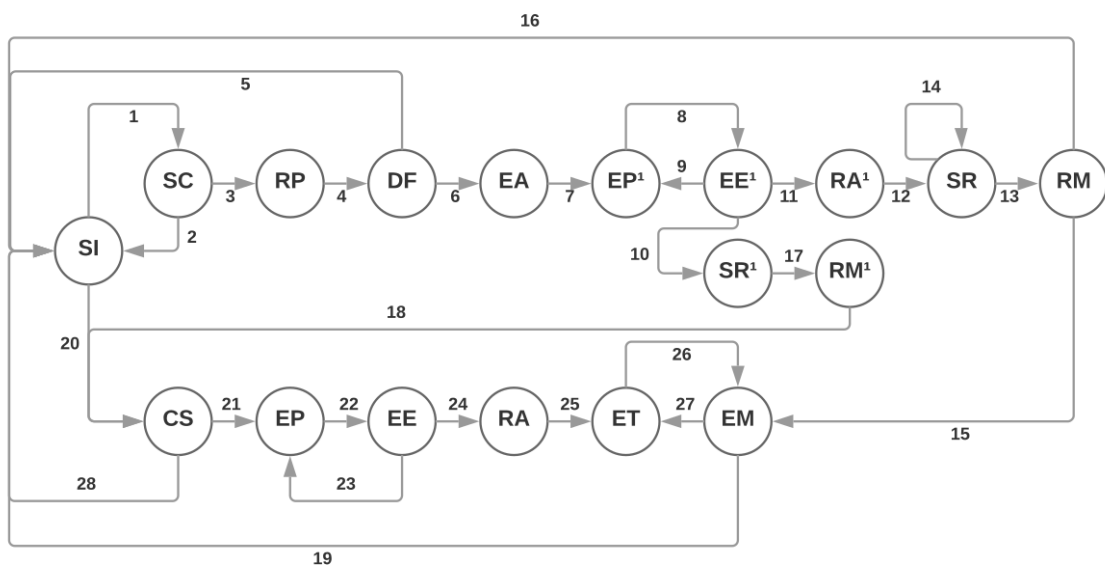
Inicialmente, o nó de origem S envia uma sequência de solicitação para enviar pacotes (*request to send* - RTS) como preâmbulos, a fim de estabelecer a comunicação com um membro do FCS (nó 1 e nó 2). No momento t_1 , o nó 1 entra no período ativo e recebe o RTS. Neste ponto, o nó 1 armazena o horário que deve acordar para receber o pacote de dados e realiza a transmissão de um pacote de confirmação (*clear to send* - CTS) como um eACK. O nó de origem S recebe essa confirmação e vai para o estado de suspensão. O nó 1 começa imediatamente com uma nova série de preâmbulos até receber um CTS do nó 3. Esse processo continua, um caminho que conecta a origem e o destino está sendo formado e o tempo de recepção do pacote está sendo agendado nesses nós. Após um tempo de espera programado, a transmissão de dados pelos nós escalonados é iniciada. Durante a transmissão do pacote de dados em uma parte da rede, os preâmbulos ainda estão sendo usados para escalonar novos nós na rede. Como exemplo,

na Figura 4.1, enquanto o nó 5 envia preâmbulos, o nó S, distante do 5, inicia a transmissão de DATA para o nó 1. A cardinalidade do FCS é cuidadosamente escolhida de modo que o tempo médio de transmissão dos preâmbulos em cada salto seja igual ao tempo de transmissão do pacote. Quando um nó envia o pacote de dados para o nó receptor e não recebe a confirmação deste em um instante de tempo t_{Ack} , o nó emissor considera como falha de comunicação. Neste caso, o nó emissor retransmite o dado após um tempo (t_{Ack}) e o nó receptor é reprogramado para receber o dado após o tempo de ($t_{Ack} + t_{Data}$) (caso ocorrido na transmissão do dado do nó 1 para o nó 3).

4.2 Máquina de estado

A máquina de estado do protocolo MAC-AHEAD é apresentada na Figura 4.2. As Tabelas 4.1 e 4.2 apresentam respectivamente os estados e as transições utilizadas na máquina de estado.

Figura 4.2 – Máquina de estado do protocolo MAC-AHEAD.



Fonte: do autor (2019)

Inicialmente, cada nó alterna entre os estados SI (SONO_INATIVO) e SC (SONDAGEM) através das transições 1 e 2. Estes estados correspondem ao ciclo de ativo/inativo do nó sensor e determinam o tempo de sondagem do canal em relação ao ciclo total.

Ao se iniciar uma nova transmissão, a partir do recebimento de mensagens das camadas superiores, o estado CS (CARRIER_SENSE) é iniciado (transição 20). Se o canal não estiver livre (transição 28), retorna para o estado SI. Se o canal estiver livre (transição 21) muda para

Tabela 4.1 – Descrição dos estados utilizados na máquina de estado

Abreviação	Estado	Descrição
SI	SONO_INATIVO	Nó está inativo
SC	SONDAGEM	Sondagem do canal p/ preâmbulo
RP	RECEBE_PRE	Preâmbulo sendo recebido
DF	DECIDE_FCS	Nó decidindo quais serão os nós do FCS
EA	ENVIO_eACK	Mensagem de confirmação sendo enviada
EP	ENVIO_PRE	Preâmbulo sendo enviado
EE	ESPERA_eACK	Sondagem do canal p/ eACK
RA	RECEBE_eACK	Recebendo eACK
SR	SONO_RECEBE	Nó está inativo até o tempo de receber a mensagem
RM	RECEBE_MSG	Dado sendo recebido
CS	CARRIER_SENSE	Sondagem do canal para atividade
ET	ESPERA_TEMPO	Nó esperando um período de tempo para enviar o dado
EM	ENVIO_MSG	Nó está enviando o dado

o estado EP (ENVIO_PRE) e transmite em *broadcast* os pacotes de preâmbulos (transição 22). Em seguida, muda para o estado EE (ESPERA_eACK). Neste estado, o protocolo MAC espera por uma resposta de um nó vizinho pertencente ao FCS. Se nenhum membro do FCS que estiver no alcance despertar e responder com um eACK, a transição 23 retorna para o estado EP. Caso um sinal de resposta seja detectado no canal (transição 24), a máquina de estados é direcionada para o estado RA (RECEBE_eACK). Ao receber a confirmação eACK (transição 25), muda para o estado ET (ESPERA_TEMPO). Neste estado, o nó calcula um tempo para acordar e receber o pacote de dados do nó anterior. A determinação deste tempo de espera é apresentada na seção de resultados. Esse tempo é necessário para evitar colisões entre os pacotes de controle (preâmbulos) e pacotes de dados. Logo em seguida, a transição 26 leva para o estado EM (ENVIO_MSG). Caso a mensagem seja enviada e perdida ao longo da transmissão (transição 27), retorna para o estado ET. Caso a mensagem seja enviada sem perda ao longo da transmissão (transição 19), retorna para o estado SI.

Em caso de não haver mensagem para ser enviada, o protocolo alterna entre os estados SI e SC. Quando um preâmbulo é detectado no estado SC, o MAC alterna para o estado RP (RECEBE_PRE) pela transição 3. Depois de receber preâmbulo de um vizinho, o estado DF é atingido pela transição 4. Neste estado, o nó que detectou o preâmbulo determina se pertence ao FCS do nó transmissor. Se este for o caso, o estado muda para EA (ESPERA_eACK) pela transição 6, caso contrário, o nó retorna para SI (pela transição 5). Após a transmissão do

pacote, o protocolo passa para o estado EP¹, no qual um preâmbulo é enviado para encontrar o próximo nó da rota.

Tabela 4.2 – Descrição das transições utilizadas na máquina de estado

Transição	Descrição
1	<i>Timeout</i> para o estado de inatividade do nó
2	<i>Timeout</i> da sondagem do canal. Nenhuma atividade detectada
3	Detectado sinal de preâmbulo
4	Preâmbulo recebido
5	Nó não é membro do FCS
6	Nó é membro do FCS
7	eACK enviado
8	Preâmbulo enviado
9	Preâmbulo não detectado Canal inativo.
10	<i>Timeout</i> colisão iminente
11	eACK detectado. Canal ativo
12	eACK recebido
13	Tempo para a recepção do dado alcançado. Dado recebido
14	Tempo para a recepção do dado alcançado. Dado não recebido
15	Nó atual não é o destino final
16	Nó atual é o destino final
17	O mesmo que 13
18	Mensagem recebida
19	Mensagem enviada sem perda
20	Mensagem pendente para a transmissão
21	Canal livre
22	Preâmbulo enviado
23	eACK não detectado
24	eACK detectado
25	eACK recebido
26	<i>Timeout</i> τ
27	Mensagem enviada com perda
28	Canal não está livre

Enviado o preâmbulo, o protocolo espera para detectar a transmissão de um eACK no estado EE¹ (transição 8). Se nenhum sinal for detectado no canal, o protocolo retornará para o EP¹. No caso em que um eACK é detectado, a transição 11 conduz o protocolo para o estado RA¹ em que eACK é recebido. Depois disso, o protocolo é direcionado para o estado SR (SONO_RECEBE) pela transição 12. Neste estado, o rádio é desativado até o instante do recebimento do pacote de dados. Quando o tempo de espera é terminado, o dado é enviado. Caso haja perda do dado enviado, o estado se mantém em SR, alcançado pela transição 14. Caso o dado seja enviado sem perda (transição 13), o estado é alterado para RM (RECEBE_MSG).

Neste estado, o pacote de dados é recebido pelo nó sensor. Após essa recepção, o endereço de destino final é verificado. Se a mensagem é endereçada ao nó receptor, o pacote é enviado para a camada superior e o protocolo retorna ao estado de repouso pela transição 16. Se o nó receptor não for o destino, a mensagem deverá ser transmitida para o próximo nó. Assim, o protocolo alcança o estado do EM pela transição 15.

Quando uma colisão iminente é identificada no estado EE^1 , a transição 10 conduz a máquina de estados ao estado SR^1 para evitar colisões. Neste estado, o nó interrompe a transmissão de preâmbulos e aguarda a chegada do pacote de dados. A transição 17 leva o protocolo para o estado RM^1 no qual a mensagem de dados é recebida. Depois disso, o nó assume o papel de iniciador e vai para o estado CS pela transição 18 e o processo de envio é reiniciado. Como os preâmbulos foram interrompidos, a transição 18 é responsável por reiniciar suas transmissões da mesma forma que foi feito no pacote de dados.

4.3 Pacotes RTS

Os pacotes RTS são os responsáveis por armazenar informações utilizadas para fazer o roteamento do protocolo MAC-AHEAD. Para o funcionamento do protocolo em cenários que apresentem buracos, os pacotes RTS se comportam como os pacotes do protocolo GPSR, de maneira a executar o modo perímetro. Desta forma, os pacotes RTSs utilizam atributos que permitem ao protocolo atuar no modo guloso e no modo perímetro.

Os atributos utilizados pelos pacotes RTS são: informação do nó emissor (*sender*), informação do nó de destino final (*finalReceiver*), lista de últimos nós do FCS (*fcs*), informação do nó de salto anterior (*previousReceiverL1*), informação de quanto tempo o dado será enviado pelo nó anterior (*dataTime*), a intensidade de sinal utilizada no envio do pacote (*signalStrength*), informação do tamanho do pacote RTS em bits (*headerLength*) e número máximo de preâmbulos a serem enviados na tentativa de encontrar nós vizinhos (*sequence*). A Figura 4.3 abaixo apresenta o pacote RTS utilizado e seus atributos.

Figura 4.3 – Formato do pacote RTS.

sender	finalReceiver	previousReceiver	
dataTime	signalStrength	headerLength	
sequence		fcs	
Lp	Lf	e0	M

Fonte: do autor (2019)

4.4 Seleção de nós do FCS

Como utiliza uma abordagem *anycast*, o protocolo MAC-AHEAD deve selecionar quais nós pertencerão ao subconjunto de nós candidatos ao próximo salto (FCS). São selecionados para compor o FCS aqueles nós que apresentam maior avanço até o nó destino e quando a qualidade do canal de comunicação para a transmissão do pacote de dados até um determinado nó for satisfatória. A qualidade satisfatória diz respeito a garantir que a entrega do dado seja com a menor latência e com o menor consumo de energia.

Para determinar quais são os nós que apresentam o maior avanço em relação ao nó de destino são utilizados algoritmos baseados no protocolo de roteamento GPSR. Estes algoritmos utilizam o modo guloso e o modo perímetro para esta finalidade. O algoritmo que permite ao protocolo atuar no modo perímetro foi adaptado para ser utilizado em protocolos assíncronos *anycast*, conforme apresentado na seção 4.6.

No MAC-AHEAD, o nó transmissor mede a intensidade de ruído (e interferência) local. Com essa informação e a informação de tamanho de pacote de dados, uma tabela é consultada pelo transmissor para saber qual é a distância limite dos vizinhos para serem incluídos no FCS. Essa distância é denominada Req.

São realizados testes para se verificar qual valor de distância Req o nó vizinho deve estar para gerar um valor de PER aceitável e que, combinados com outros valores de parâmetros do protocolo, tal como tempo de espera para a envio do dado e o tamanho do pacote de dado transmitido, permita que a entrega do dado seja com menor latência e com menor consumo de energia.

Estas verificações são realizadas pelo nó emissor para cada um dos nós vizinhos. Serão selecionados para compor o FCS aqueles nós vizinhos que satisfazem estas verificações relacionadas ao roteamento e a qualidade do canal.

4.5 Heurística para determinar a cardinalidade do FCS

A cardinalidade do FCS no protocolo MAC-AHEAD é importante para controlar a velocidade da emissão dos preâmbulos transmitidos. Este controle de velocidade é importante para que não haja colisões entre dados e preâmbulos.

Como descrito na subseção 3.2.2, o protocolo PAX-MAC utiliza uma quantidade de nós no FCS fixa para cada um dos diferentes tamanhos de pacotes transmitidos. Com o intuito de

reduzir o tempo de latência para a entrega do dado desenvolveu-se uma nova heurística para a escolha da quantidade de nós no FCS. Esta quantidade impacta na latência pelo fato desta ser responsável pela utilização de um número maior ou menor de hops durante o processo de entrega do dado.

A nova heurística implementada leva em consideração a medida de avanço médio. Esta medida diz qual é o avanço médio de um dado sendo transmitido de um nó X para um nó de destino Y considerando diferentes números de nós no FCS. Para obter esta medida, calcula-se a média da distância entre o nó atual e cada um dos seus vizinhos que apresentam maior avanço, considerando 1,2,3,4,5,6 e 7 nós no FCS.

A heurística consiste em obter a melhor relação entre avanço médio e avanço médio teórico. Inicialmente, escolhe-se a quantidade de nós no FCS a ser utilizado como o padrão C_p . A cada novo salto, calcula-se a relação entre avanço médio e avanço médio teórico para C_{p-1} , C_p e C_{p+1} nós no FCS. A relação que apresentar o maior valor será a cardinalidade utilizada para o FCS.

Esta heurística tem por objetivo retirar o nó do FCS que esteja distante dos nós que apresentam maior avanço, utilizando a cardinalidade C_{p-1} ou inserir um nó no FCS que ofereça um avanço próximo aos nós que oferecem um maior avanço, usando cardinalidade C_{p+1} .

4.6 Protocolo MAC-AHEAD: roteamento no modo perímetro para comunicação *anycast* com preâmbulos avançados

Quando o protocolo MAC-AHEAD encontra um buraco (ponto de mínimo) no terreno, ele passa a realizar o roteamento no modo perímetro. Com intuito de permitir que o protocolo MAC-AHEAD entregue dados em regiões que contenham buracos, foi adicionado um algoritmo baseado no modo perímetro do protocolo GPSR. A seguir são apresentadas as etapas utilizadas para o perfeito funcionamento do algoritmo que permite ao protocolo utilizar o modo perímetro adaptado para comunicação *anycast* com preâmbulos avançados.

4.6.1 Fluxo de pacotes RTS: roteamento

O roteamento realizado pelo protocolo MAC-AHEAD utiliza os pacotes RTS e algoritmos que escolhem o próximo nó através do modo guloso ou do modo perímetro.

Inicialmente um pacote RTS é criado possuindo o atributo modo (M) como sendo guloso. Em seguida, um algoritmo define a quantidade de nós (v) que fará parte do FCS e realiza-se o roteamento. O roteamento é realizado conforme o modo que o pacote RTS está atualmente.

Estando no modo guloso, o algoritmo de roteamento verifica as distâncias entre a localização de cada um dos seus nós vizinhos até a localização do nó de destino final. Havendo os v nós vizinhos, aqueles que possuírem a menor distância, sendo portanto responsável pelo maior avanço, serão escolhidos para fazer parte do FCS. Este algoritmo de roteamento será executado até a entrega do dado ao destino final, caso não haja falha.

Caso não haja nenhum dos nós vizinhos que satisfaça este critério (casos em que o próprio nó está mais próximo do nó destino do que seus vizinhos - buracos), o pacote RTS terá alguns atributos modificados. O primeiro atributo modificado será o M , que deixa de ter o valor de guloso e passa a ter o valor de perímetro. O atributo Lp é inicializado com o valor da localização do nó atual, que fez com que o protocolo entrasse no modo perímetro. Em seguida, o algoritmo descrito a seguir realiza o roteamento no modo perímetro, selecionando quais nós farão parte do FCS. Após escolhido um dos nós do FCS a encaminhar o dado, o atributo Lf e $e0$ recebe a referência do nó atual e do nó a ser encaminhado o dado.

Quando o nó receptor recebe o pacote RTS, estas informações do pacote são armazenadas na classe *MacAheadState*, responsável por armazenar informações temporariamente entre um estado da MAC e outro. Logo em seguida, o pacote RTS é criado, contendo as informações armazenadas em *MacAheadState* e realiza novamente todo o fluxo descrito de acordo com o modo que ele esteja, guloso ou perímetro.

4.6.2 Algoritmo para a escolha de nós utilizando comunicação *anycast* em cenários com buraco

Quando o pacote RTS está no modo perímetro, existe a necessidade de executar um algoritmo de roteamento que contorne buracos. O protocolo GPSR no modo perímetro utiliza um algoritmo que realiza a regra da mão direita para este objetivo, com roteamento *unicast*. Entretanto, para roteamentos *anycast* com transmissão assíncrona, um outro algoritmo foi desenvolvido. Este novo algoritmo utiliza a regra da mão direita combinada com limiar de ângulo para a escolha de outros nós que irão compor o subconjunto de possíveis nós do próximo salto.

Inicialmente, este novo algoritmo realiza a regra da mão direita. Obtêm-se o ângulo entre duas arestas. Logo que ocorre a falha, este algoritmo obtém o ângulo entre aresta 1,

formado pelo nó atual e nó de destino final e aresta 2, formada por nó atual e nó de cada um dos seus vizinhos. A partir do segundo nó, no modo perímetro, o algoritmo obtém o ângulo entre aresta 1, formado pelo nó atual e nó anterior e aresta 2, formada por nó atual e nó de cada um dos seus vizinhos.

O par formado por número do nó e ângulo são armazenados em uma lista *smallerAngles*. Logo em seguida, esta lista é ordenada em forma crescente de ângulos e o primeiro nó da lista é definido como primeiro nó do subconjunto.

A partir daí é definido um limite de ângulo que será utilizado para permitir que outros nós vizinhos façam parte do subconjunto, a partir do primeiro nó escolhido. A utilização de um limiar angular foi de extrema importância para garantir que os nós escolhidos para compor o subconjunto permitam que o pacote de dados esteja de fato contornando a região com buraco.

O valor de limiar angular foi escolhido após experimentos realizados. Nestes experimentos, o protocolo foi executado para entregar dado de um nó de origem para um nó de destino. Três valores diferentes de limiares angulares foram utilizados, sendo eles: 10, 15 e 20 graus. O valor de limiar angular de 10 graus permitiu com que o protocolo entregasse dados com o menor tempo e com maior taxa de entrega. Para os valores de 15 e 20 graus percebeu-se que os nós escolhidos estavam desviando o dado do contorno do buraco, aumentando o número de hops utilizados para a entrega e conseqüentemente a latência.

A Tabela 4.3 apresenta a relação das latências obtidas na entrega de dado de tamanho de 15% do tempo de ciclo com um valor de Req de 28m utilizando diferentes ângulos. Os parâmetros utilizados para este teste são os parâmetros apresentados na Tabela 5.1, no Capítulo 5.

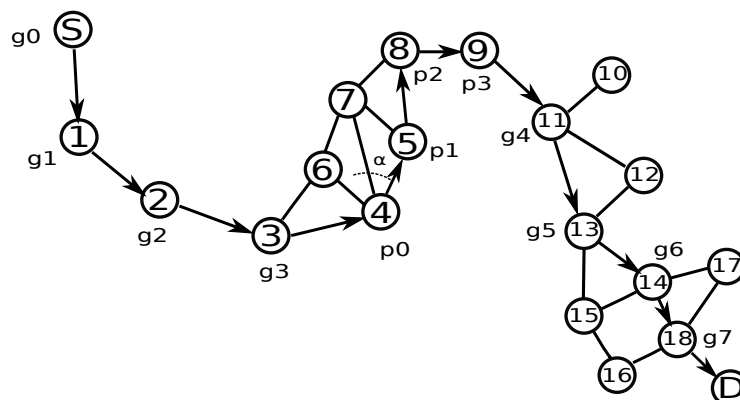
Tabela 4.3 – Latência obtida na entrega de dados de tamanho de 15% do tempo de ciclo com um Req de 28 m. A menor latência deu-se com um ângulo de 10°

Ângulo (°)	Latência (duração de pacote)
10	59,60
15	72,83
20	102,34

Para a escolha dos outros nós da lista *smallerAngles* que irão compor o subconjunto, foram escolhidos os nós que possuem um ângulo menor ou igual ao ângulo formado pelo ângulo do primeiro nó escolhido somado ao ângulo limite escolhido (10 graus). Quando a quantidade de nós que satisfazem esta condição é maior do que o número de elementos do subconjunto escolhido, os nós que apresentam maior avanço tem prioridade.

A Figura 4.4 apresenta a atuação do algoritmo de modo perímetro implementado. A partir de um ponto de mínimo local (nó 4), são selecionados os nós que estão dentro de um intervalo angular α (nós 5 e 7).

Figura 4.4 – Seleção de nós do FCS a partir de um mínimo local (nó 4). Nós 5 e 7 são selecionados por estarem dentro de um intervalo angular α



Fonte: do autor (2019)

4.6.3 Problemas no desempenho do MAC-AHEAD ao se utilizar roteamento no modo perímetro

Alguns problemas foram identificados e consequentemente solucionados para a melhor performance e eficiência do protocolo MAC-HEAD ao se utilizar o roteamento no modo perímetro. A seguir são apresentados os problemas ocorridos e a descrição da solução implementada:

a) Nós do FCS com direção oposta ao contorno do buraco

Um dos problemas observados diz respeito a seleção dos nós pertencentes ao FCS em caso de se utilizar o algoritmo no modo perímetro. Pode-se observar que houve momentos em que os nós selecionados para fazer parte do FCS foram nós que estavam no sentido oposto ao contorno do buraco.

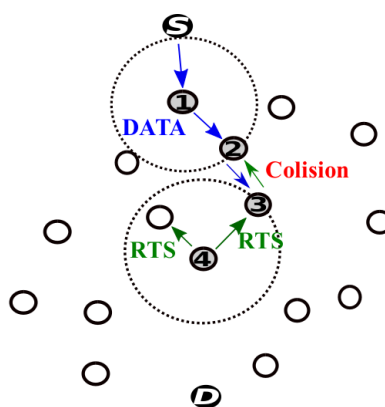
Solução: Para solucionar este problema, um outro atributo foi adicionado ao pacote RTS contendo as informações dos últimos nós do FCS escolhidos. Após isto, um outro critério adicionado para a escolha dos nós do FCS foi que estes nós não poderiam ter sido escolhidos anteriormente (com exceção de quando havia somente nós do FCS já utilizados

para serem escolhidos). Com esta solução, os nós escolhidos anteriormente para compor o FCS erroneamente, deixaram de ser escolhidos.

b) Alto índice de colisões entre pacote de dados e mensagens RTS ao contornar o buraco

Outro problema verificado foi a grande quantidade de colisões que ocorrem entre pacote de dados e mensagens RTS. Como as mensagens RTS estão sempre a frente com o intuito de descobrir a rota, ocorre que ao contornar o buraco, as mensagens RTS podem estar no sentido de contorno do buraco (sentido para cima na figura) enquanto o pacote de dado está sendo ainda encaminhado pela rota descoberta por elas (sentido para baixo). Isto faz com que um grande número de colisões ocorra. Este problema é apresentado na Figura 4.5.

Figura 4.5 – Cenário apresentando as colisões entre mensagens de dados (direção pra baixo) e mensagens RTS (direção pra cima) ocorridas ao contornar buracos.



Fonte: do autor (2019)

Solução: Para solucionar este problema, fez-se com que o protocolo MAC-AHEAD, quando entrasse no modo perímetro, recebesse o dado, de onde ele estivesse, antes de iniciar o contorno do buraco com a descoberta de rota pelas mensagens RTS. Este comportamento é similar a quando o protocolo detecta uma colisão.

c) Detecção de colisões iminentes

Inicialmente, o protocolo MAC-AHEAD utiliza um sistema de detecção de colisões utilizado pelo protocolo PAX-MAC. Este sistema de detecção de colisões permite que o nó detecte colisões para até dois hops atrás deste nó. Este sistema é ineficiente, pois caso o pacote de dado esteja a N hops atrás do nó que envia mensagens RTS, sendo $N > 2$, no mesmo alcance de transmissão do rádio, este sistema não detecta a colisão. Um outro problema encontrado diz respeito a esta questão.

Solução: A fim de solucionar este problema um novo sistema de detecção de colisões foi implementado. Este novo sistema considera o alcance de transmissão do nó que está enviando o pacote de dado e o tempo que este nó vai enviar o dado. Para isto é criado um novo atributo no pacote RTS que armazena a informação do tempo de envio do dado para cada um dos 6 últimos nós do caminho. Sendo assim, antes de começar o envio de uma mensagem RTS é verificado qual nó mais distante destes 6 está no mesmo alcance de transmissão do nó que envia a mensagem RTS e se o tempo de envio do dado é igual ao tempo de envio da mensagem RTS. Caso isto ocorra, é detectado colisão.

A partir desta novo sistema de detecção de colisões, o protocolo MAC-AHEAD passou a contornar eficientemente os buracos testados.

4.7 Resumo das características dos protocolos MAC-AHEAD e PAX-MAC

A Tabela 4.4 apresenta de forma resumida as características dos protocolos MAC-AHEAD e PAX-MAC. Dessa maneira é possível observar as contribuições do protocolo MAC-AHEAD.

Tabela 4.4 – Características dos protocolos MAC-AHEAD e PAX-MAC

Característica	MAC-AHEAD	PAX-MAC
Assíncrono	sim	sim
<i>Anycast</i>	sim	sim
<i>cross-layer</i>	sim	sim
Utiliza <i>duty-cycle</i>	sim	sim
Detecta dados perdidos por ruído ou interferência de sinal	sim	não
Retransmite dados perdidos por ruído ou interferência de sinal	sim	não
Seleciona nós do FCS que estejam a melhor distância	sim	não
Seleciona nós do FCS de acordo com a qualidade do canal de comunicação	sim	não
Transmite diferentes tamanhos de dados	sim	sim
Permite transmitir dados em regiões com a presença de mínimos locais (buracos)	sim	não
Utiliza cardinalidade do FCS fixa	não	sim

A seguir são apresentadas as configurações utilizadas nos experimentos.

5 CONFIGURAÇÃO DOS EXPERIMENTOS

O desempenho do MAC-AHEAD foi avaliado em simulação com o simulador Grubix Wireless Sensor Networks, um redesenho e extensão do ShoX (LESSMANN; HEIMFARTH; JANACIK, 2008). Um modelo de rádio baseado no IEEE 802.15.4 foi empregado com potência de transmissão fixa, links bidirecionais e modelo de propagação de espaço livre para fonte pontual isotrópica.

Para detectar a perda do pacote de dados transmitido no simulador Grubix, utilizou-se as equações definidas pelo modelo referência IEEE 802.15.4 para o cálculo da taxa de erros de bits (*Bit Error Rate* - BER) e para o cálculo da taxa de erros de pacotes (*Packet Error Rate* - PER). A Equação 5.1 foi utilizada para o cálculo do BER (IEEE, 2003):

$$BER_{802.15.4} = \frac{8}{15} \times \frac{1}{16} \times \sum_{k=2}^{16} -1^k \binom{16}{k} e^{(20 \times SINR \times (\frac{1}{k} - 1))} \quad (5.1)$$

Para o cálculo do PER foi utilizada a Equação 5.2, apresentada em (IEEE, 2003):

$$PER = 1 - (1 - BER)^n \quad (5.2)$$

sendo n o tamanho total do pacote de dado em bits e BER a taxa de erros de bits.

O modelo de canal de comunicação utilizado foi o Canal Binário Simétrico (*Binary Symmetric Channel* - BSC) (COVER; THOMAS, 2006).

O cenário era uma rede densa com nós estáticos espalhados aleatoriamente em uma área quadrada de 500m x 500m, de modo que suas localizações formassem um processo de *Poisson*. O grafo de conexão resultante é chamado de grafo geométrico aleatório (LI et al., 2009). Cada nó conhece sua própria posição e as posições de seus vizinhos, já que eles trocam informações de posição logo após a implantação.

Todos os nós são capazes de medir a intensidade do sinal recebido, a fim de calcular o ruído e interferência nos sinais localmente, bem como a força do sinal de comunicação. O ciclo de trabalho foi o mesmo para todos os nós com intervalo igual entre dois tempos ativos consecutivos, mas não houve sincronização em seu tempo de ativação.

Para avaliar o desempenho de MAC-AHEAD, foi comparado com os seguintes protocolos existentes: X-MAC (BUETTNER et al., 2006) e PAX-MAC (HEIMFARTH et al., 2016). A potência de transmissão utilizada foi de $P_{Tx} = 1mW$ (0dBm). Um valor de SINR padrão de 0dB foi considerado para receptores distantes de 30 m (d_0) de transmissores, o que é com-

patível com medições reais feitas com os trajetos de MicaZ (Ali; Driberg; Sebastian, 2011), transmitindo pacotes de 240 bits de comprimento. Utilizando esses parâmetros, foram obtidos uma taxa de erro de pacote $PER = 2\%$. Para diferentes comprimentos de pacotes, diferentes distâncias devem ser definidas entre o transmissor e o receptor, para atingir o mesmo PER sob as mesmas condições de ruído e interferência. Essa distância é denominada alcance de rádio equivalente (Req). Por exemplo, um $Req = 28m$ é alcançado para um pacote de 1250 bits, para se obter o mesmo $PER=2\%$. A Equação 5.3 foi utilizada para cálculo do SNR:

$$SINR = SINR_0 \left(\frac{d_0}{d}\right)^2 \quad (5.3)$$

Para inclusão de membros no FCS, o MAC-AHEAD leva em consideração o Req relacionado ao transmissor. O mesmo procedimento não está presente no PAX-MAC nem no X-MAC. Em simulações com ambos os protocolos, foi considerado um modelo único de comunicação de disco de um Req específico. Para a apresentação dos resultados, o valor da latência foi normalizada para o tempo de duração do pacote.

Os parâmetros utilizados na configuração dos experimentos são apresentados na Tabela 5.1.

Tabela 5.1 – Parâmetros utilizados na simulação experimental

Parâmetro	Valor
Alcance do rádio equivalente R_{eq} (m)	26, 28 e 30
Potência dissipada pelo nó P(mW)	60
Distância do nó fonte p/ o nó destino (m)	650
Densidade dos nós ($\frac{nodes}{m^2}$)	$8,0 \times 10^{-3}$
tempo de ciclo (s)	0.1
Tempo duração de pacote (% tempo de ciclo)	15, 20, 25, 33, 50
Número máximo de preâmbulos	98
Duração do preâmbulo (s)	$0,512 \times 10^{-3}$
Tempo de sondagem do canal (s)	$1,024 \times 10^{-3}$
Tempo de espera inicial (duração de pacotes)	3, 4, 5, 6, 7, 8, 9 e 10
SINR0	0db
d_0 (m)	30

6 RESULTADOS E DISCUSSÕES

Este capítulo apresenta análises e discussões sobre os resultados obtidos, a partir das configurações anteriormente apresentadas.

6.1 Definição do melhor tempo de espera e Req aceitável

Uma importante métrica para o funcionamento do protocolo MAC-AHEAD é o intervalo de tempo necessário para que o nó de origem comece a transmitir o dado. Este tempo é importante para que não haja colisões entre o dado que está sendo enviado e as mensagens RTS que estão sendo transmitidas à frente para encontrar a rota e sincronizar os nós.

Outra métrica de extrema importância, que contribui como sendo um fator adicional deste protocolo em relação aos outros existentes é o valor da Req que permite ter um valor de PER aceitável, de maneira que o protocolo entregue o dado com uma menor latência e com menor consumo de energia.

A fim de se obter o melhor tempo de espera para que o nó de origem libere o dado e o melhor valor de Req, uma análise experimental foi realizada. Neste experimento, foram transmitidos pacotes de diferentes tamanhos (15%, 20%, 25%, 33%, 50% do tempo de ciclo), com valores de Req diferentes (26, 28 e 30 m). Para esta análise, foram considerados erros durante a transmissão e utilizada a nova heurística para a escolha do número de elementos no FCS. A região utilizada foi a definida na configuração de experimentos.

O Quadro 6.1 abaixo apresenta os melhores parâmetros obtidos para o protocolo MAC-AHEAD. Neste quadro, é apresentado também a média do número pacotes de dados perdidos (PER) e o número médio de colisões iminentes ocorridas em cada transmissão (Colisões).

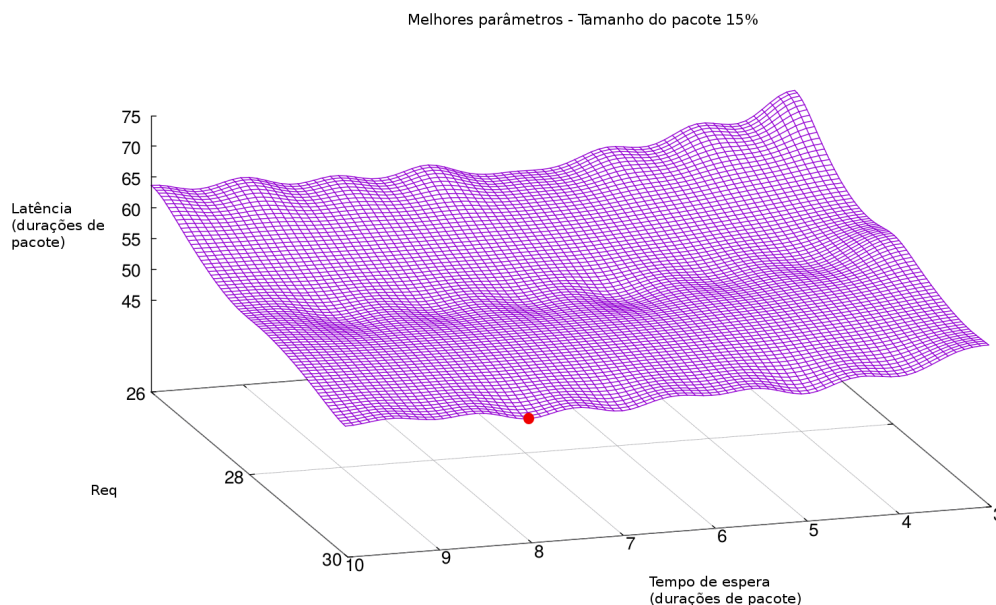
Quadro 6.1 – Parâmetros que resultaram em uma menor latência para o protocolo MAC-AHEAD quando realizada a transmissão do dado.

Tamanho do pacote (% tempo de ciclo)	Tempo de espera (duração de pacote)	Req (m)	Latência (duração de pacote)	PER	Colisões
15	8	30	49.83	3.3	0.21
20	7	30	46.6	4.66	0.35
25	8	28	44.27	1.23	0.25
33	6	28	40.84	1.98	0.37
50	6	28	40.30	3.34	0.33

Conforme apresentado no Quadro 6.1 acima, a melhor combinação de parâmetros a ser utilizada para a transmissão de um pacote de 15% do ciclo total de tempo é um tempo de espera de 8 durações de pacote e um valor de Req de 30 m. A Figura 6.1 apresenta as latências obtidas

na transmissão destes pacotes. O ponto apresenta a combinação que possibilitou que o pacote de dado fosse entregue com a menor latência.

Figura 6.1 – Latências obtidas na transmissão do dado com tamanho de 15% do tempo de ciclo, para diferentes tempos de espera e valores de Reqs. O ponto destaca a combinação que permitiu que o pacote fosse entregue com a menor latência.



Fonte: do autor (2019)

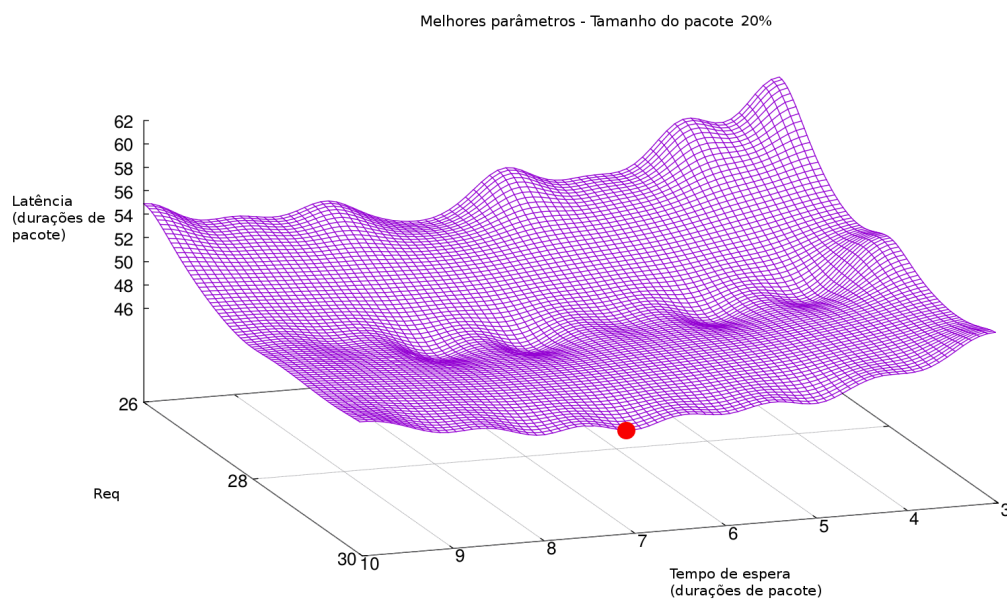
Pela Figura 6.1, observa-se que as maiores latências foram obtidas quando utilizado valor de Req 26 m combinado com um tempo de espera menor (3 e 4 durações de pacote). Isto se deve ao fato de que para menores tempos de espera, o número de colisões é maior entre dados e mensagens RTS, até que seja encontrado um tempo de espera ótimo. Após encontrado um tempo de espera ótimo, o valor da latência volta a aumentar pelo fato de se utilizar tempos de espera superiores ao esperado para reduzir as colisões.

A menor latência obtida quando utilizado um valor de Req de 30m deve-se ao menor número de colisões ocorridas em comparação com outros parâmetros para o mesmo tamanho de pacote, embora os valores de PER sejam altos para este mesmo valor de Req.

Para a transmissão de um pacote de dado com tamanho de 20% do tempo de ciclo, deve-se utilizar um tempo de espera de 7 durações de pacote a um valor de Req de 30m a fim de obter-se a menor latência (46.6 tempos de durações de pacote). A Figura 6.2 apresenta uma relação entre as latências obtidas para a transmissão deste pacote quando utilizado diferentes tempo de espera e valores de Req.

A Figura 6.2 apresenta um comportamento semelhante ao da Figura 6.1, uma vez que as maiores latências obtidas foram registradas para valores de Req de 26m e menores tempos de espera. Observa-se que, para menores tempos de espera, as latências são altas e seguem decrescendo até encontrar um valor ótimo. Após encontrado o tempo ótimo, as latências passam a aumentar em relação a menor latência encontrada.

Figura 6.2 – Latências obtidas na transmissão do dado com tamanho de 20% do tempo de ciclo. para diferentes tempos de espera e Reqs. O ponto destaca a combinação que permitiu que o pacote fosse entregue com a menor latência.



Fonte: do autor (2019)

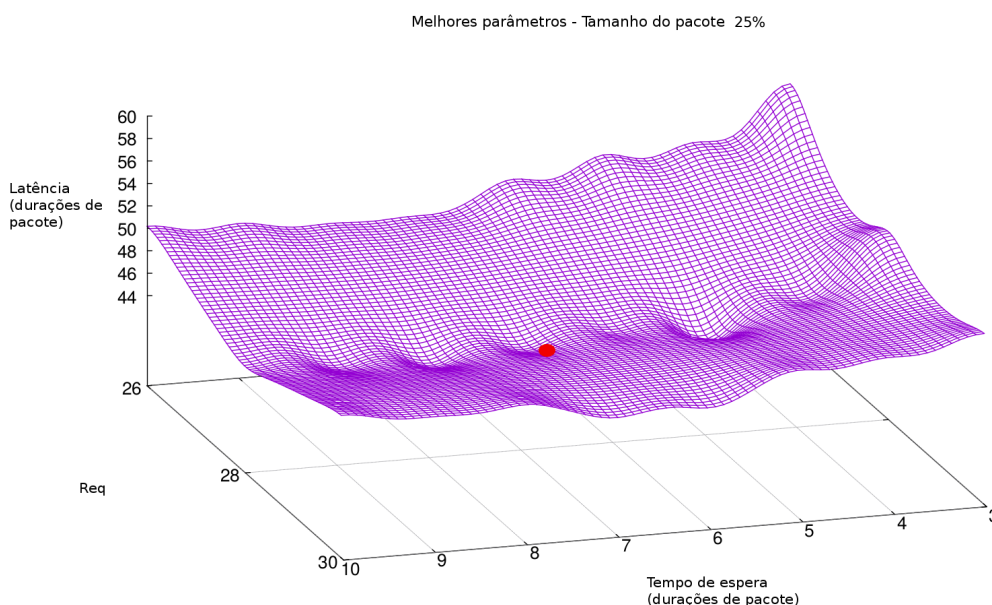
A menor latência encontrada com a combinação ótima de tempo de espera e valor de Req gerou um valor alto de PER e baixos valores de colisões, quando comparados com outras combinações de tempo de espera e Req. O alto valor de PER deve-se ao fato do valor de SINR ser menor quando utilizado grandes distâncias, conforme apresentado na Equação 5.3. Valores menores de SINR geram um valor de BER maior e consequentemente um maior valor de PER.

A Figura 6.3 apresenta a relação das latências obtidas para a transmissão do pacote de tamanho de 25% do tempo de ciclo utilizando diferentes tempos de espera e valores de Req. Conforme observado, a melhor combinação de parâmetros encontrados a fim de transmitir este pacote com uma menor latência (44.27 tempos de durações de pacote) é utilizar um tempo de espera de 8 durações de pacote e um valor de Req de 28 m, destacado com um ponto na Figura.

Observa-se que para a transmissão do pacote de 25% o valor de PER foi menor quando comparado com a transmissão de pacotes de 15% e 20% com valores de Req de 30m. Este fato

comprova mais uma vez a relação entre o alto valor de SINR para Reqs menores o que implica em menores valores de BER e consequentemente de PER.

Figura 6.3 – Latências obtidas na transmissão do dado com tamanho de 25% do tempo de ciclo. para diferentes tempos de espera e Reqs. O ponto destaca a combinação que permitiu que o pacote fosse entregue com a menor latência.



Fonte: do autor (2019)

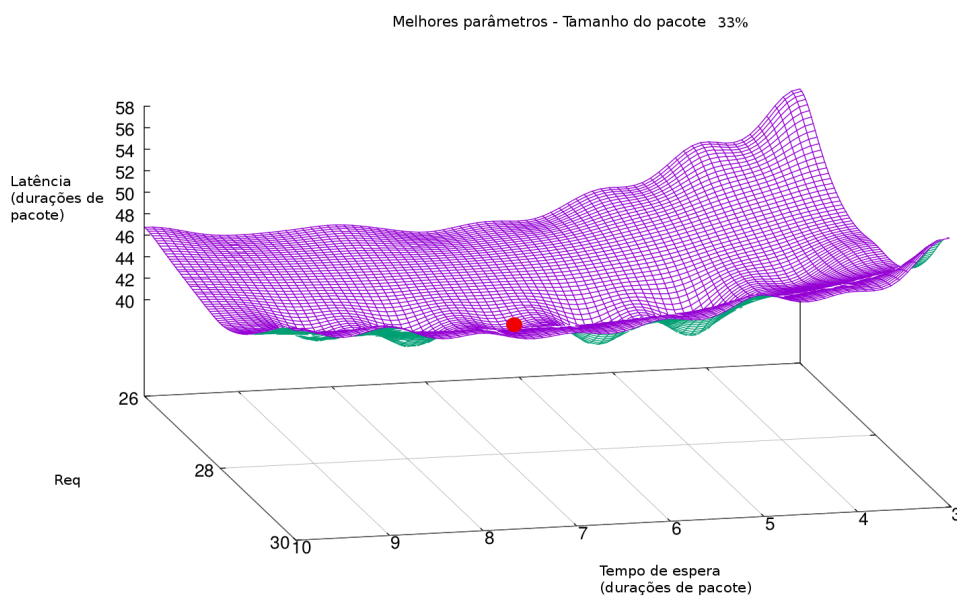
Pelo Quadro 6.1, observa-se que a menor latência foi obtida com parâmetros que permitiram ao protocolo entregar o dado com valores menores de PER, como discutido, e também com um número menor de colisões, quando comparado com transmissões de tamanhos de pacotes de 20%.

Para a transmissão de pacotes com tamanho de 33% do ciclo total do tempo, deve-se utilizar um tempo de espera de 6 durações de pacote e um Req de 28 metros, a fim de transmitir com a menor latência (40.84 tempos de durações de pacote). A Figura 6.4 apresenta uma relação entre os diferentes tempos de espera e os diferentes valores de Req utilizados para a transmissão de pacotes com tamanho de 33% do ciclo total de tempo, e as latências obtidas. O ponto representa a combinação que permitiu obter a menor latência.

Pode-se observar pelo Quadro 6.1 que, para este cenário, a latência foi menor em relação a transmissão de pacotes com tamanho de 15%, 20% e 25% do ciclo total do tempo. Este menor valor de latência foi possível por utilizar-se um valor de Req (28m) menor do que o utilizado na transmissão de pacotes com tamanho de 15% e 20% (30m) e por ter ocorrido menor perda de pacotes em relação também a estes dois tamanhos de pacotes transmitidos. A menor latência

em relação a transmissão de um pacote com tamanho de 25% do ciclo total de tempo deve-se ao fato de que, para a transmissão de pacotes com tamanho de 33% é utilizado um número menor de nós no FCS (média de 2 nós), enquanto que para tamanhos de pacote de 25% utiliza-se uma média de 3 nós no FCS. Um número menor de nós no FCS garante que o protocolo avance com maiores distâncias a cada hop, reduzindo assim o número de hops e consequentemente o valor da latência.

Figura 6.4 – Latências obtidas na transmissão do dado com tamanho de 33% do tempo de ciclo. para diferentes tempos de espera e valores de Reqs. O ponto destaca a combinação que permitiu que o pacote fosse entregue com a menor latência.



Fonte: do autor (2019)

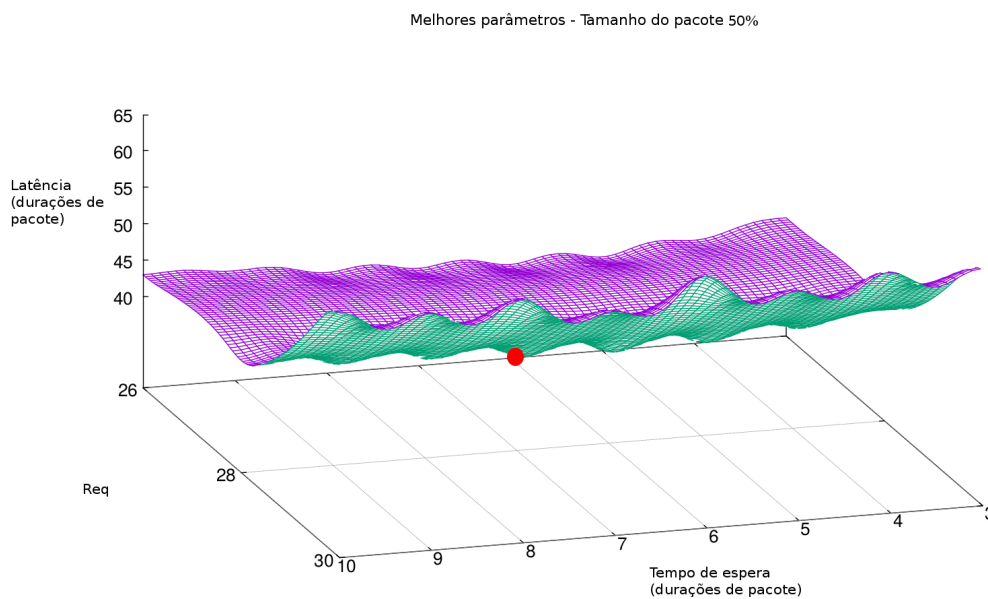
Para a transmissão de pacotes com tamanho de 50% do ciclo total do tempo deve-se utilizar um tempo de espera de 6 durações de pacote e um valor de Req de 28 metros, a fim de se transmitir com a menor latência (40.30 tempos de durações de pacote). A Figura 6.5 apresenta uma relação entre os diferentes tempos de espera, os diferentes valores de Req utilizados para a transmissão destes pacotes e as latências obtidas. O ponto representa a combinação que permitiu obter-se a menor latência.

Pode-se perceber que para a transmissão deste pacote obteve-se uma latência menor do que todas as latências obtidas na transmissão dos pacotes anteriores com tamanhos de 15%, 20%, 25% e 33% do ciclo total do tempo. Isto se deve ao menor valor de hops utilizados para a entrega, quando comparado com pacotes de 15% e 25%; ao menor valor de PER encontrado,

quando comparado com pacotes de 20% e ao menor número de colisões obtidas, quando comparado com pacotes de 33%.

Outro comportamento diferente em relação as transmissões de pacotes anteriores, foi as maiores latências concentradas em cenários que utilizam um Req de 30m. Ou seja, a combinação feita entre alto número de bits transmitidos a um maior valor de Req faz com que o pacote seja transmitido com alta latência.

Figura 6.5 – Latências obtidas na transmissão do dado com tamanho de 50% do tempo de ciclo. para diferentes tempos de espera e valores de Reqs. O ponto destaca a combinação que permitiu que o pacote fosse entregue com a menor latência.



Fonte: do autor (2019)

6.2 Heurística para escolha do número de nós do FCS - Desempenho

Para analisar o desempenho da nova heurística utilizada para determinar o número de nós inseridos no FCS foi realizado uma comparação entre o protocolo PAX-MAC, que possui um número de nós no FCS fixo e o MAC-AHEAD, que apresenta um número de nós no FCS variável.

Para esta comparação foi realizada a transmissão de dados de diferentes tamanhos (15%, 20%, 25%, 33% e 50% do tempo de ciclo) de um nó de origem para um nó de destino, com uma distância de 650 metros entre eles. O protocolo PAX-MAC utilizado na comparação foi o original que não detecta erros nem realiza retransmissão. Para o protocolo PAX-MAC foram utilizado os melhores parâmetros de tempo de espera. Para o protocolo MAC-AHEAD foram

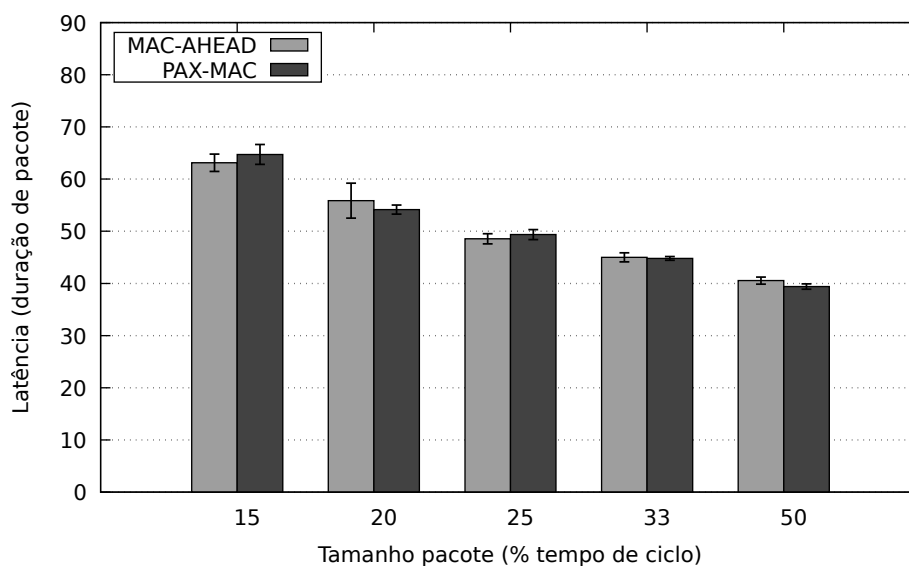
utilizados os parâmetros (tempo de espera e Req) que permitiram ao protocolo ter o menor valor de PER. Estes parâmetros são apresentados no Quadro 6.2.

Quadro 6.2 – Parâmetros que permitiram ao protocolo ter a menor taxa de erros de pacotes durante a transmissão do dado.

Tamanho do pacote (% tempo de ciclo)	Tempo de espera (duração de pacote)	Req (m)	PER
15	6	26	0.09
20	7	26	0.07
25	8	26	0.14
33	7	26	0.2
50	6	26	0.42

O desempenho de cada um dos protocolos quando transmitidos os dados de diferentes tamanhos podem ser observados na Figura 6.6 que apresenta as diferentes latências para cada um dos dados de diferentes tamanhos transmitidos:

Figura 6.6 – Latências obtidas na transmissão de diferentes tamanhos de pacote de dado, para os protocolos MAC-AHEAD e PAX-MAC. MAC-AHEAD utiliza um tamanho de FCS variável



Fonte: do autor (2019)

Conforme observado, as diferenças entre cada uma das latências resultantes para cada um dos protocolos durante a entrega dos dados de um mesmo tamanho foram mínimas, ou seja, as diferenças ficaram dentro do intervalo de confiança.

A utilização de uma nova heurística para o protocolo MAC-AHEAD para a escolha do número de nós no FCS permitiu que ele entregasse dados a uma menor latência para os pacotes

de tamanho 15% e 25% apenas. Para a transmissão de dados de tamanhos 20%, 33% e 50% o protocolo PAX-MAC entregou o dado com um tempo menor.

O motivo encontrado para esta heurística não ter tido o desempenho esperado foi que ao longo da transmissão os nós inseridos no FCS ora apresentavam maior avanço e ora não apresentavam tanto avanço assim. Isto pode ocorrer caso um maior número de nós do FCS esteja a uma distância maior e para trás dos nós que apresentam maior avanço.

6.3 Protocolos MAC-AHEAD, X-MAC e PAX-MAC - Cenário sem buracos

Para a validação do mecanismo de detecção de erros e de retransmissão implementado foi realizado um teste comparando a performance do protocolo MAC-AHEAD com a performance de outros dois importantes protocolos do estado da arte, sendo eles: o X-MAC e o PAX-MAC. Para isto, foram considerados perdas de pacotes de dados ao longo da transmissão.

Para esta comparação foram realizadas simulações para a transmissão de diferentes tamanhos de pacotes (15%, 20%, 25%, 33% e 50% do tempo de ciclo). Os parâmetros gerais utilizados foram os apresentados no Quadro 5.1, para ambos os protocolos. Afim de obter-se o melhor desempenho de todos os protocolos foram utilizados os parâmetros ótimos para cada um deles.

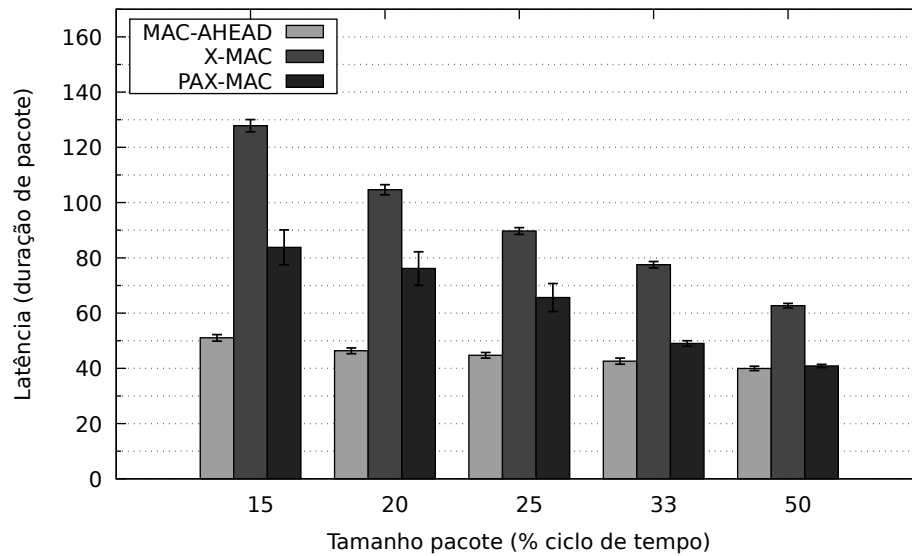
Para o protocolo MAC-AHEAD, o tempo de espera e melhor valor de Req utilizados foram os apresentados na Quadro 6.1, pois equivalem aos parâmetros ótimos encontrados.

Para o protocolo PAX-MAC foi utilizado o tempo de espera ótimo para cada tamanho de pacote transmitido, descrito em Heimfarth et al. (2016) com um valor de Req de 28 metros. Este valor de Req foi utilizada tanto para o protocolo PAX-MAC quanto para o protocolo X-MAC. O motivo de ter escolhido este valor é por que foi o menor, dentre todas a outros testados, a gerar parâmetros ótimos para o protocolo MAC-AHEAD e por não haver até o momento estudos que relacionem melhores valores de Req a serem aplicados em ambos os protocolos.

6.3.1 Análise comparativa da Latência

A Figura 6.7 apresenta as latências obtidas para a transmissão de diferentes tamanhos de pacotes de dado. Conforme observado, o protocolo MAC-AHEAD foi capaz de entregar o dado ao nó destino com uma latência menor do que o protocolo X-MAC e PAX-MAC para todos os tamanhos de pacotes transmitidos.

Figura 6.7 – Latências obtidas na transmissão de diferentes tamanhos de pacote de dado.



Fonte: do autor (2019)

Observa-se que a diferença de latência entre os protocolos MAC-AHEAD e X-MAC para cada tamanho de pacote diminui a medida que se transmite um pacote de tamanho maior, conforme apresentado no Quadro 6.3.

Quadro 6.3 – Diferença entre a maior e a menor latência para a entrega do dado pelo protocolo X-MAC e MAC-AHEAD, para os diferentes tamanhos de pacote.

Tamanho do pacote (% tempo de ciclo)	Diferença entre Latências (duração de pacote)
15	76,825
20	58,332
25	44,98
33	34,905
50	22,724

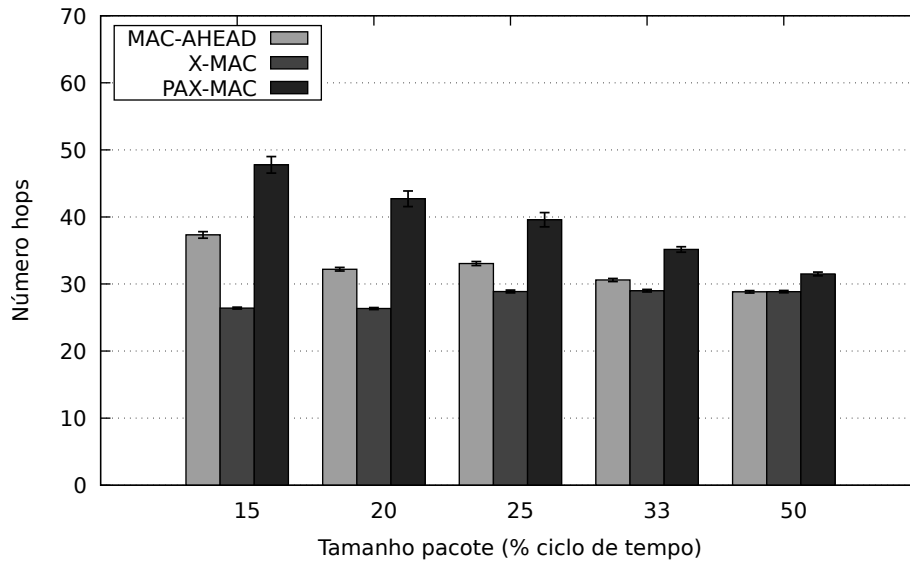
Isto se deve ao fato de que, com o aumento do tamanho de pacote, um número menor de elementos no FCS é utilizado. Desta forma, para pacotes cujo tamanho é de 50% do tempo de ciclo, o número de nós no FCS pode ser em média um, o que iguala ao comportamento *unicast* do protocolo X-MAC.

A menor latência para a entrega dos pacotes de dados realizada pelo protocolo MAC-AHEAD em comparação com o protocolo X-MAC, deve-se a sua característica de roteamento *anycast*. Isto se deve ao fato de que a probabilidade de um nó sensor acordar antes ser maior para um conjunto maior de nós, o que agiliza o processo de encontrar qual nó será o próximo encaminhador. Por outro lado, isto pode acarretar em um número maior de hops utilizados

para a entrega do dado, visto que nem sempre os nós com maior avanço são escolhidos para encaminhá-lo.

A Figura 6.8 apresenta o número de hops médios utilizados para a entrega dos diferentes tamanhos de pacotes de dado para cada um dos protocolos.

Figura 6.8 – Número de hops utilizados para a transmissão de diferentes tamanhos de pacotes de dado.



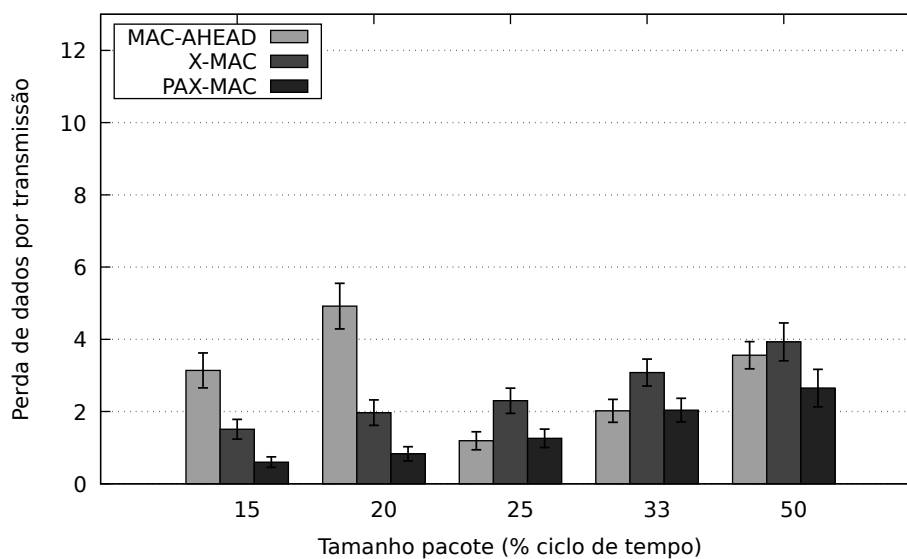
Fonte: do autor (2019)

Conforme observado, o número de hops utilizados para a entrega de dados pelo protocolo MAC-AHEAD é maior do que o utilizado para a entrega do dado pelo protocolo X-MAC. A diferença do número de hops utilizado pelos dois protocolos para entregar o dado é reduzida a zero a medida que aumenta o tamanho do pacote transmitido. Isto se deve ao mesmo motivo apresentado anteriormente.

A menor latência para a entrega do dado pelo protocolo MAC-AHEAD quando comparado ao protocolo PAX-MAC deve-se a característica do protocolo MAC-AHEAD de poder se adaptar ao Req em cenários com canal que apresenta ruído e interferência. Ou seja, para cenários que apresentem erros, este utiliza de melhores parâmetros. A melhor latência não é necessariamente acompanhada de valores baixos de perda de dados como pode ser observado na Figura 6.9.

Percebe-se pela Figura 6.9 que, para os pacotes de tamanho 15% e 20% do tempo de ciclo transmitido, o valor de perda de pacotes do protocolo MAC-AHEAD é bem maior quando comparado ao protocolo PAX-MAC. Isto se deve ao fato do valor de Req utilizado no protocolo MAC-AHEAD ser 30m, neste caso, enquanto o valor de Req do protocolo PAX-MAC ser 28m.

Figura 6.9 – Média de Perda de dados por transmissão de diferentes tamanhos de pacote de dado.



Fonte: do autor (2019)

Para os outros tamanhos de pacotes transmitidos os valores de perda de pacotes são aproximados pois utilizam o mesmo valor de Req (28m).

A utilização de melhores parâmetros no protocolo MAC-AHEAD permitiu ao protocolo ter valores de número de colisões reduzidos. A Figura 6.10 abaixo apresenta o valor do número de colisões de cada um dos protocolos. Desta forma, este motivo pode ser considerado como o principal fator para a menor latência deste protocolo em relação ao protocolo PAX-MAC, para os pacotes transmitidos de tamanhos 15%, 20%, 25%, 33% e 50%. Para a transmissão do pacote de tamanho 50%, a menor latência é justificada também pelo menor valor de hops utilizados para a entrega.

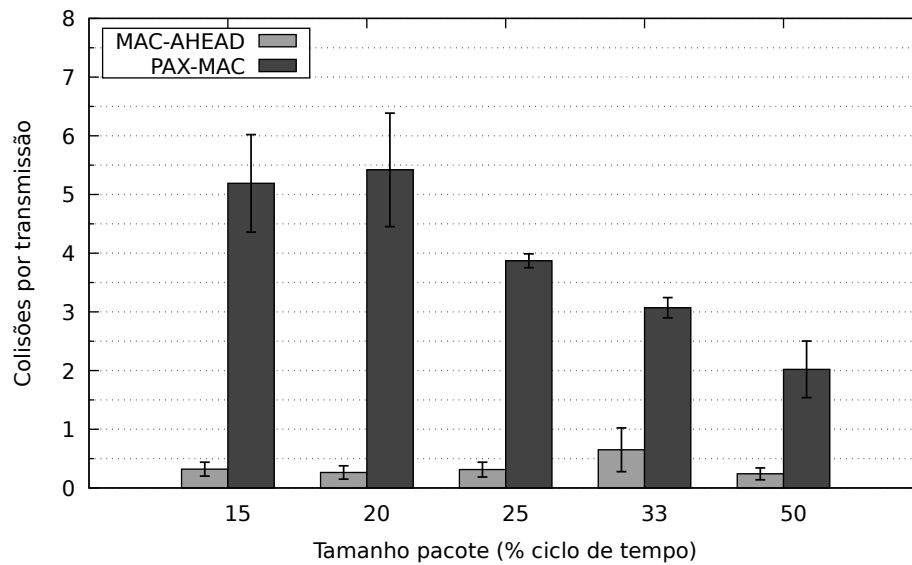
O menor número de colisões é devido ao tempo de espera ótimo utilizado, que permite o protocolo liberar o dado em um momento que reduza a proximidade com as mensagens RTSs.

6.3.2 Análise comparativa do consumo energético

O consumo energético resultante da transmissão dos pacotes de dados pelos diferentes protocolos pode ser observado na Figura 6.11.

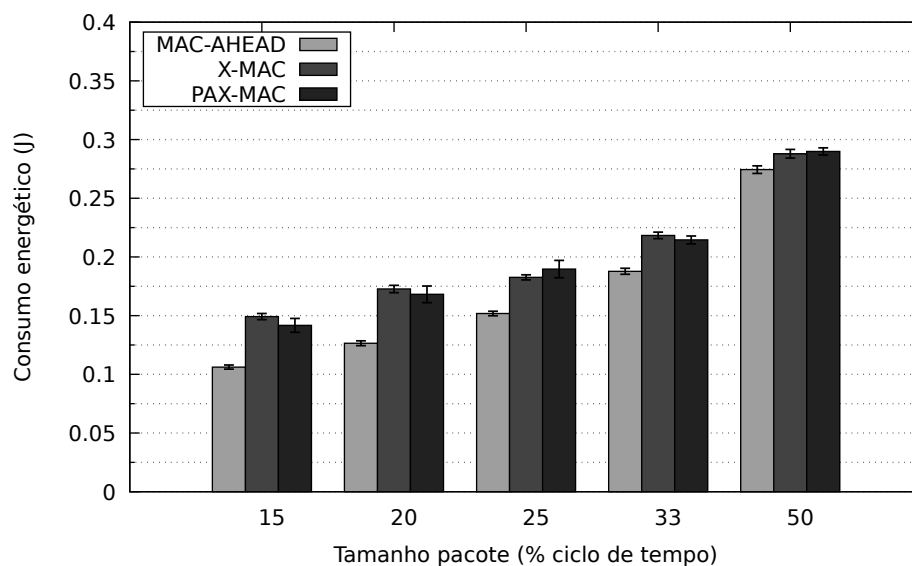
Como pode ser observado, o protocolo MAC-AHEAD foi capaz de entregar o dado de um nó de origem para um nó de destino, tendo um gasto energético menor do que os outros dois protocolos.

Figura 6.10 – Porcentagem média de colisões ao longo da transmissão de diferentes tamanhos de pacotes de dado.



Fonte: do autor (2019)

Figura 6.11 – Consumos energéticos resultantes da transmissão de diferentes tamanhos de pacote de dado.



Fonte: do autor (2019)

O menor consumo energético do protocolo MAC-AHEAD em relação ao protocolo X-MAC está inerentemente relacionado a melhor latência resultante na entrega de dados. Como discutido anteriormente, pelo fato do MAC-AHEAD possuir um roteamento *anycast* a probabilidade de um nó encontrar o seu vizinho acordado é maior, o que reduz a quantidade de mensagens RTS transmitidas e o consequente consumo de energia relacionado a estas transmissões. É possível observar que mesmo apresentando um menor valor de PER, para a transmissão

de pacotes de tamanho 15% e 20%, o protocolo X-MAC ainda assim consumiu mais energia. Ou seja, embora tenha ocorrido menos retransmissões nestes dois casos, o maior número de mensagens RTSs transmitidas, em comparação com a quantidade transmitida pelo protocolo MAC-AHEAD, elevou o valor do consumo de energia.

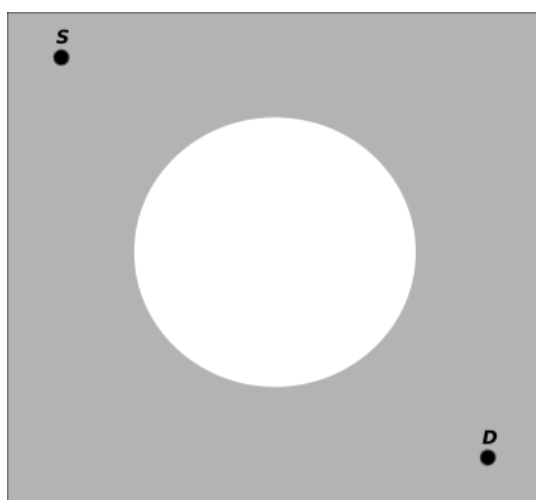
O menor consumo energético do protocolo MAC-AHEAD em relação ao protocolo PAX-MAC deve-se a dois fatores, sendo eles: menor número de colisões ocorridas na transmissão dos pacotes de tamanhos 15%, 20%, 25%, 33% e 50%, maior número de hops utilizados na entrega do dado com pacote de tamanho 50%.

6.4 Procolos MAC-AHEAD, X-MAC e PAX-MAC - Cenário com buracos

Para a validação do protocolo MAC-AHEAD em cenários que apresentem buracos, dois cenários foram utilizados.

No cenário 1, utiliza-se um buraco em formato circular no centro do terreno de dimensões 500x500. O nó de origem é fixado no ponto (0,15) e o nó de destino no ponto (475,475), gerando uma distância de 650 metros. O terreno utilizado neste cenário pode ser observado na Figura 6.12. O nó indicado pela letra S equivale ao nó de origem e o nó indicado pela letra D equivale ao nó de destino. A região de cor cinza equivale a região onde os outros nós foram espalhados e a região de cor branca equivale ao buraco (região sem nós).

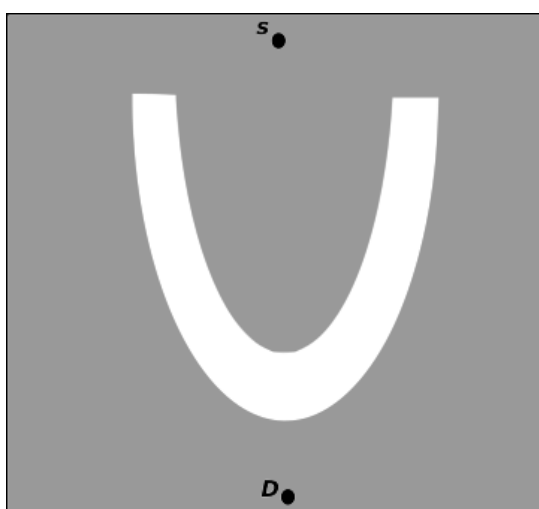
Figura 6.12 – Cenário 1 apresentando um buraco circular no terreno. A região de cor cinza representa a região onde os outros nós foram espalhados e a região de cor branca representa a região do buraco.



Fonte: do autor (2019)

No cenário 2, utiliza-se um buraco no formato da letra U. Neste cenário, o terreno possui dimensões 650mx650m. O nó de origem é fixado no ponto (150,0) e o nó de destino no ponto (150,650), gerando uma distância de 650 metros. O terreno utilizado neste cenário pode ser observado na Figura 6.13. O nó indicado pela letra S equivale ao nó de origem e o nó indicado pela letra D equivale ao nó de destino. A região de cor cinza representa a região onde os outros nós foram espalhados e a região de cor branca representa a região sem nós.

Figura 6.13 – Cenário 2 apresentando um buraco no terreno em formato da letra U. A região de cor cinza representa a região onde os outros nós foram espalhados e a região de cor branca representa a região do buraco.



Fonte: do autor (2019)

Para o protocolo X-MAC foi utilizado o protocolo de roteamento geográfico GPSR padrão. O protocolo PAX-MAC utilizado corresponde ao original, ou seja, possui apenas o roteamento GPSR no modo guloso, sem a característica de contornar buracos.

A seguir são apresentados os resultados obtidos quando executados os três protocolos nestes dois cenários.

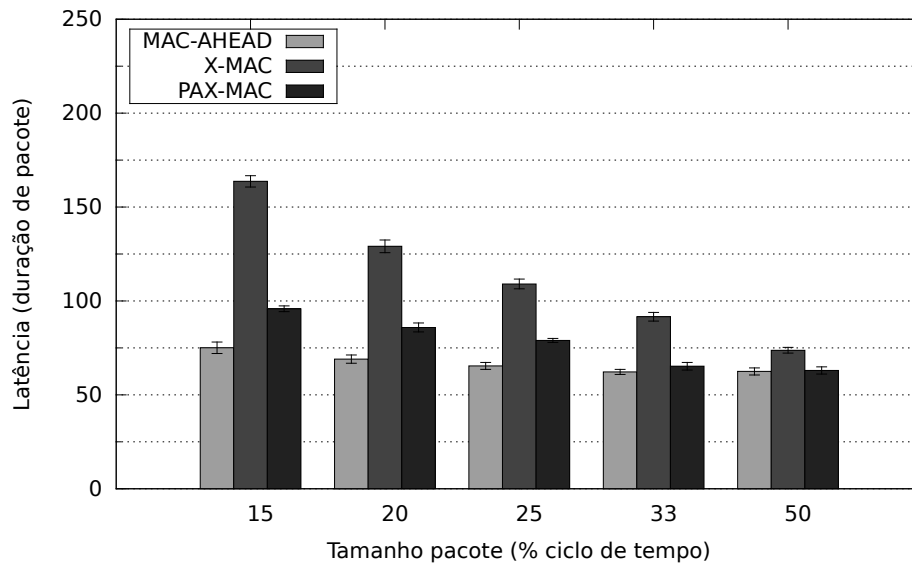
6.4.1 Cenário 1 - Análise comparativa

Para o cenário 1, foi realizada a comparação entre os três protocolos, MAC-AHEAD, X-MAC e PAX-MAC. Este cenário possui a característica de permitir que protocolos MAC que não utilizam protocolos de roteamento para contornar buraco possam realizar a entrega do dado na maior parte das vezes. Porém, há momentos que este cenário apresenta mínimos locais.

- Latência

As latências resultantes da entrega de diferentes tamanhos de pacotes de dado para cada um dos três protocolos utilizando o cenário 1 podem ser observadas na Figura 6.14:

Figura 6.14 – Latência resultante da transmissão de pacotes de dados de tamanhos diferentes para cada um dos três protocolos no cenário 1.



Fonte: do autor (2019)

Conforme observado, a implementação de algoritmos baseados no protocolo de roteamento GPSR que contorne buracos para o protocolo assíncrono e *anycast* MAC-AHEAD, fez com que este mantivesse o seu desempenho, dado que foi possível entregar o dado a uma menor latência mesmo em terrenos com buraco que apresente o cenário 1.

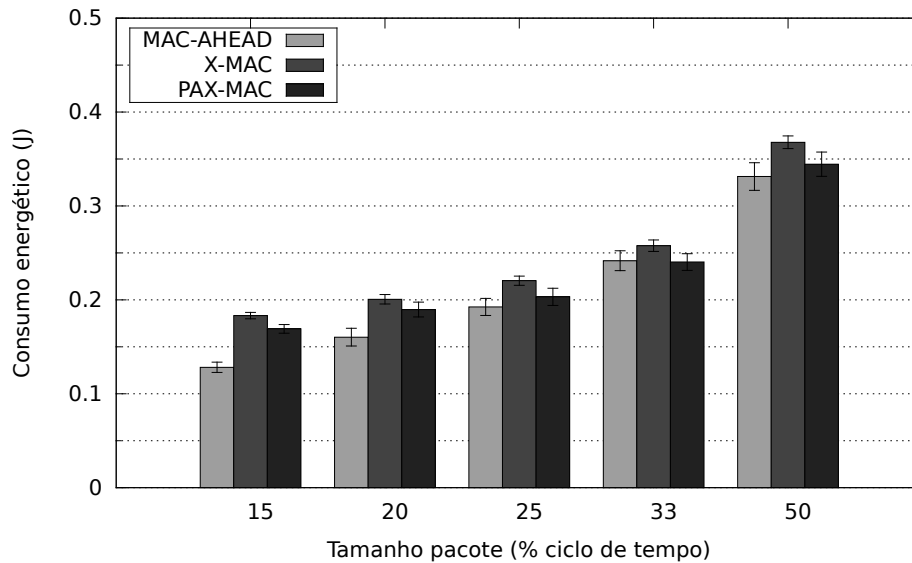
Semelhante ao apresentado na Figura 6.7, o comportamento da latência resultante da entrega do dado pelo protocolo MAC-AHEAD em comparação com o protocolo X-MAC manteve-se o mesmo, ou seja, a diferença entre elas diminuíam para a transmissão de pacotes de tamanhos maiores de dado. Isto se justifica pelo número de elementos no FCS que tendem a ficar próximos para a transmissão de pacotes de tamanhos maiores.

Comparando as latências resultantes da entrega do dado pelos protocolos MAC-AHEAD e PAX-MAC, percebe-se que as diferenças entre elas foram maiores para os tamanhos de pacotes 15% 20% e 25%. A menor latência foi justificada pelo menor número de colisões ocorridas quando entregue o pacote de dado pelo protocolo MAC-AHEAD. Este menor número de colisões pode ser justificado pelo fato dos parâmetros de tempo de espera inicial do protocolo MAC-AHEAD considerarem erros no canal de transmissão e portanto serem mais realistas.

- Consumo energético

Conforme apresentado na Figura 6.15 o protocolo MAC-AHEAD conseguiu entregar o dado de diferentes tamanhos com um consumo energético menor, mesmo em terrenos com buraco que apresente a características do cenário 1.

Figura 6.15 – Consumo energético resultante da transmissão de pacotes de dado de tamanhos diferentes para cada um dos três protocolos no cenário 1.



Fonte: do autor (2019)

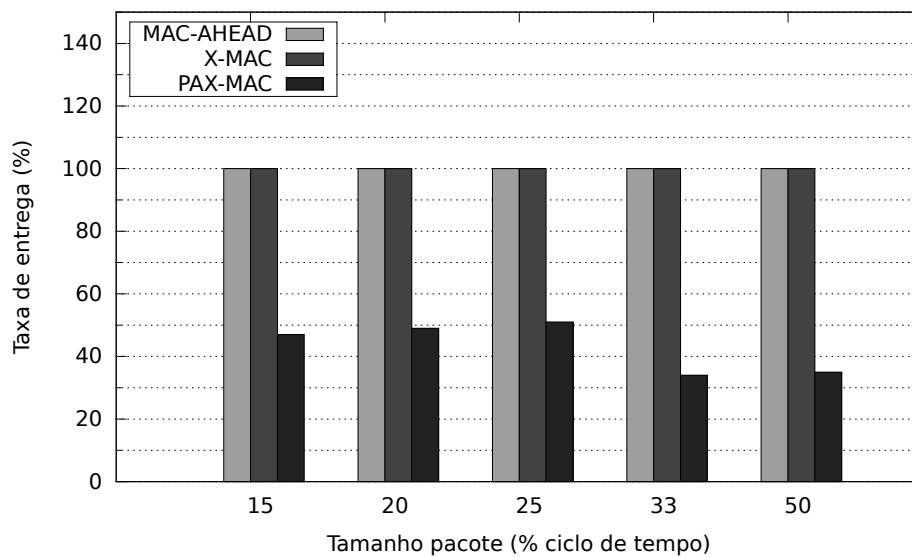
Por utilizar um protocolo de roteamento *anycast*, o protocolo MAC-AHEAD possui vantagens quanto ao número de mensagens RTSs enviadas para encontrar o próximo nó, em comparação com o protocolo X-MAC que possui um roteamento *unicast*. Esta vantagem diz respeito ao menor número de mensagens RTSs que devem ser enviadas para encontrar os nós pertencentes a um grupo e que estão acordados, ao contrário de enviar uma mensagem RTS para um único nó esperando o momento em que ele irá acordar. Esta abordagem impacta diretamente no menor consumo de energia dos protocolos MAC-AHEAD e PAX-MAC em relação ao protocolo X-MAC, quando realizado a transmissão do dado.

Os consumos de energia do protocolo MAC-AHEAD em relação ao protocolo PAX-MAC foram menores para os pacotes de tamanhos 15%, 20%, 25% e 50%. Isto de certa forma esta atrelado a justificativa da menor latência obtida por este protocolo, ou seja, um menor número de colisões ocorridas.

- Taxa de Entrega

A taxa de entrega de pacotes de dados de diferentes tamanhos realizado pelos três protocolos quando utilizando o cenário 1 pode ser observado na Figura 6.16. Neste cenário, o protocolo PAX-MAC não teve 100% da taxa de entrega para a transmissão dos pacotes. Isto pode ser justificado pelo fato deste terreno, parte das vezes em que foi gerado, apresentar nós que geram pontos de falha na atuação do modo guloso do roteamento (ponto de mínimo).

Figura 6.16 – Taxa de entrega dos pacotes de dados transmitidos, utilizando o cenário 1.



Fonte: do autor (2019)

6.4.2 Cenário 2 - Análise comparativa

O cenário 2 possui a particularidade de permitir que para que haja entrega de dado o protocolo deva utilizar um protocolo de roteamento que permita contornar buracos. Desta forma, embora tenha sido realizado testes com o protocolo PAX-MAC neste cenário, não houve entrega de dado por parte dele. A seguir são apresentados os resultados quando comparados os protocolos MAC-AHEAD e X-MAC.

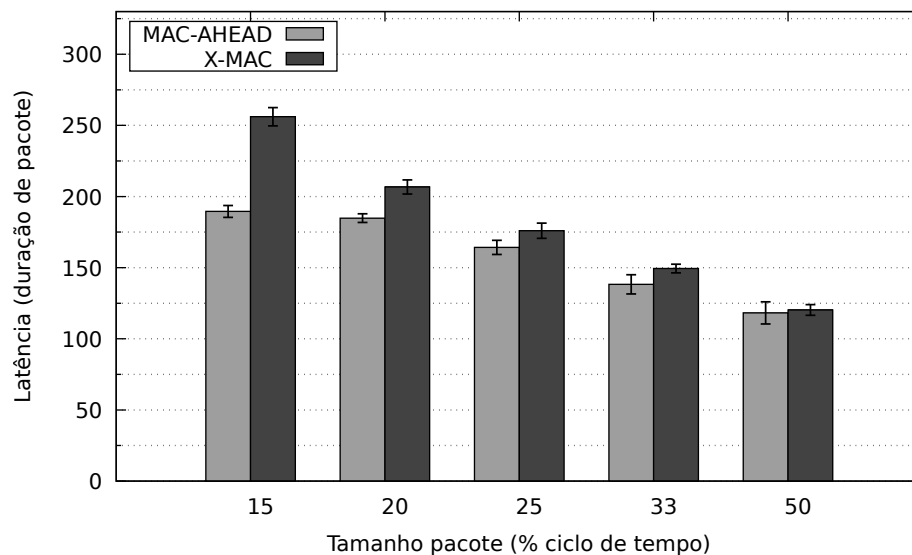
- Latência

As latências resultantes da entrega de pacotes de tamanhos diferentes de dado para cada um dos dois protocolos, utilizando o cenário 2 pode ser observado na Figura 6.17. Conforme apresentado, observa-se que o protocolo MAC-AHEAD entregou o dado de um nó de origem para um nó de destino com uma latência menor do que o protocolos X-MAC.

Para o tamanho de pacote de 50% do tempo de ciclo os valores de latências ficaram próximos.

As menores latências em relação ao protocolo X-MAC deve-se pela característica de possuir um roteamento *anycast*, ao contrário do X-MAC que possui roteamento *unicast*. A diferença entre as latências deste dois protocolos vai diminuindo conforme o tamanho do pacote transmitido aumenta pelo fato de que para pacotes maiores um número menor nós no FCS é utilizado se igualando a característica *unicast*.

Figura 6.17 – Latência resultante da transmissão de pacotes de dados de tamanhos diferentes para cada um dos dois protocolos no cenário 2.

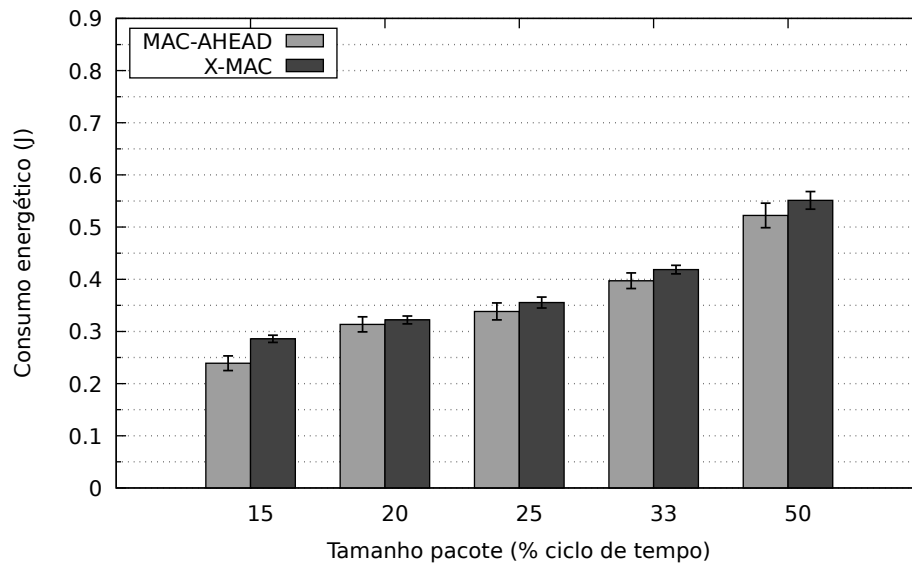


Fonte: do autor (2019)

- Consumo energético

O consumo energético resultante da entrega de diferentes tamanhos de pacotes de dados pelos dois protocolos utilizando o cenário 2 é apresentado na Figura 6.18. Pode-se perceber que o protocolo MAC-AHEAD entregou os dados de diferentes tamanhos com um consumo energético menor. Como discutido anteriormente, isto se deve ao fato de que o protocolo X-MAC é *unicast* e utiliza um número maior de mensagens RTS para encontrar um nó acordado. O protocolo MAC-AHEAD, por ser *anycast*, utiliza um número maior de nós no FCS e acaba enviando um número menor de mensagens RTS. Isto aumenta a probabilidade de um nó acordar mais rápido.

Figura 6.18 – Consumo energético resultante da transmissão de pacotes de dados de tamanhos diferentes para cada um dos dois protocolos no cenário 2.



Fonte: do autor (2019)

7 CONSIDERAÇÕES FINAIS

Este projeto apresentou o protocolo MAC-AHEAD, um protocolo *cross-layer* que integra funcionalidades da camada de rede e da camada MAC. Este protocolo utiliza o método de *duty-cycle* para reduzir o consumo de energia dos nós sensores e realiza roteamento *anycast*. A entrega de pacotes de dados é possível a partir das trocas de mensagens de controle RTS, que escalona os nós definindo o tempo em que estes devem estar acordados para receber o pacote de dados.

O número de nós que compõe o FCS é variável e definido a partir de uma nova heurística desenvolvida. Esta heurística visa considerar como membros FCS apenas os nós que garantem um maior avanço em relação ao nó de destino a partir de uma média de avanço teórico previamente calculada.

O protocolo MAC-AHEAD apresenta um mecanismo de detecção de perda de pacote de dados. Este mecanismo permite ao protocolo identificar se houve perda de pacote ao transmiti-lo. A partir da detecção da perda de pacote é possível retransmiti-lo com o mecanismo de retransmissão desenvolvido. Desta forma, o protocolo reduz a perda de pacotes.

Outra característica adicional é a capacidade de contornar regiões com buracos. Esta característica foi alcançada através da adaptação do protocolo GPSR no modo perímetro para funcionar com *anycast* e adaptar-se ao MAC-AHEAD.

Como fator diferencial de outros protocolos existentes, o protocolo MAC-AHEAD possui a característica de selecionar os nós do FCS baseado na qualidade do canal de comunicação e na distância que os nós vizinhos estão do nó emissor.

Os resultados de simulações foram utilizados para identificar, a partir do tamanho do pacote de dados, qual o melhor tempo de espera e qual o valor de Req que resulta em um menor valor de latência e consumo de energia.

O protocolo MAC-AHEAD foi comparado com outros dois protocolos, o X-MAC e o PAX-MAC em cenários com regiões de buraco e sem regiões de buraco. A comparação demonstrou que MAC AHEAD entregou pacotes de dado a uma menor latência e com menor consumo energético para os cenários testados.

Como trabalhos futuros, pretende-se fazer com que o protocolo MAC-AHEAD possa realizar transmissões em multi canais, expandindo sua atuação.

REFERÊNCIAS

- AHMED, I.; PENG, M.; WANG, W. Energy efficient cooperative nodes selection in wireless sensor networks. In: **2007 International Conference on Parallel Processing Workshops (ICPPW 2007)**. [S.l.: s.n.], 2007. p. 50–50. ISSN 0190-3918.
- AKYILDIZ, I. et al. Wireless sensor networks: a survey. **Computer Networks**, v. 38, n. 4, p. 393 – 422, 2002. ISSN 1389-1286. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1389128601003024>>.
- AL-KARAKI, J. N.; KAMAL, A. E. Routing techniques in wireless sensor networks: a survey. **IEEE Wireless Communications**, v. 11, n. 6, p. 6–28, Dec 2004. ISSN 1536-1284.
- Ali, N. A.; Drieberg, M.; Sebastian, P. Deployment of micaz mote for wireless sensor network applications. In: **2011 IEEE International Conference on Computer Applications and Industrial Electronics (ICCAIE)**. [S.l.: s.n.], 2011. p. 303–308.
- ALVI, A. N. et al. Enhanced tdma based mac protocol for adaptive data control in wireless sensor networks. **Journal of Communications and Networks**, v. 17, n. 3, p. 247–255, June 2015. ISSN 1229-2370.
- 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**. [S.l.: s.n.], 2011. p. 19–27. ISSN 2155-5486.
- BACHIR, A. et al. Mac essentials for wireless sensor networks. **IEEE Communications Surveys Tutorials**, v. 12, n. 2, p. 222–248, Second 2010. ISSN 1553-877X.
- Beevi, M. J. A fair survey on internet of things (iot). In: **2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS)**. [S.l.: s.n.], 2016. p. 1–6.
- BUETTNER, M. et al. X-mac: A short preamble mac protocol for duty-cycled wireless sensor networks. In: **Proceedings of the 4th International Conference on Embedded Networked Sensor Systems**. New York, NY, USA: ACM, 2006. (SenSys '06), p. 307–320. ISBN 1-59593-343-3. Disponível em: <<http://doi.acm.org/10.1145/1182807.1182838>>.
- BULUSU, N. et al. Scalable coordination for wireless sensor networks: self-configuring localization systems. In: **International Symposium on Communication Theory and Applications (ISCTA 2001), Ambleside, UK**. [S.l.: s.n.], 2001.
- CANO, C. et al. Low energy operation in wsns: A survey of preamble sampling {MAC} protocols. **Computer Networks**, v. 55, n. 15, p. 3351 – 3363, 2011. ISSN 1389-1286. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1389128611002349>>.
- COVER, T. M.; THOMAS, J. A. **Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)**. New York, NY, USA: Wiley-Interscience, 2006. 184-191 p. ISBN 0471241954.
- 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**. [S.l.: s.n.], 2007. p. 1478–1486. ISSN 0743-166X.

EL-HOYDI, A.; DECOTIGNIE, J.-D. Wisemac: An ultra low power mac protocol for multihop wireless sensor networks. In: **Proceedings of the 1st International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS)**. Berlin: Springer-Verlag, 2004. p. 18–31.

GANESAN, D. et al. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. **SIGMOBILE Mob. Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 5, n. 4, p. 11–25, out. 2001. ISSN 1559-1662. Disponível em: <<http://doi.acm.org/10.1145/509506.509514>>.

GOYAL, R.; KHANDNOR, P. Routing protocols based on sink mobility in wireless sensor networks #x2014; a review. In: **2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)**. [S.l.: s.n.], 2017. p. 2131–2135.

HANDY, M. J.; HAASE, M.; TIMMERMANN, D. Low energy adaptive clustering hierarchy with deterministic cluster-head selection. In: **4th International Workshop on Mobile and Wireless Communications Network**. [S.l.: s.n.], 2002. p. 368–372.

HEIMFARTH, T.; GIACOMIN, J. C.; ARAUJO, J. P. d. Aga-mac: Adaptive geographic anycast mac protocol for wireless sensor networks. In: **2015 IEEE 29th International Conference on Advanced Information Networking and Applications**. [S.l.: s.n.], 2015. p. 373–381. ISSN 1550-445X.

HEIMFARTH, T. et al. A preamble ahead anycast protocol for wsns. In: **2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)**. [S.l.: s.n.], 2016. p. 458–466. ISSN 1550-445X.

HEINZELMAN, W. B.; CHANDRAKASAN, A. P.; BALAKRISHNAN, H. An application-specific protocol architecture for wireless microsensor networks. **IEEE Transactions on Wireless Communications**, v. 1, n. 4, p. 660–670, Oct 2002. ISSN 1536-1276.

HOBLOS, G.; STAROSWIECKI, M.; AITOUCHE, A. Optimal design of fault tolerant sensor networks. In: **Proceedings of the 2000. IEEE International Conference on Control Applications. Conference Proceedings (Cat. No.00CH37162)**. [S.l.: s.n.], 2000. p. 467–472.

HONG, S.-H.; KIM, H. ki. A multi-hop reservation method for end-to-end latency performance improvement in asynchronous mac-based wireless sensor networks. **Consumer Electronics, IEEE Transactions on**, v. 55, n. 3, p. 1214–1220, August 2009. ISSN 0098-3063.

IEEE. Ieee std.802.15.4 standard. 2003.

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. Disponível em: <<http://doi.acm.org/10.1145/345910.345953>>.

KAUR, T.; KUMAR, D. Tdma-based mac protocols for wireless sensor networks: A survey and comparative analysis. In: **2016 5th International Conference on Wireless Networks and Embedded Systems (WECON)**. [S.l.: s.n.], 2016. p. 1–6.

KIM, J. et al. Minimizing delay and maximizing lifetime for wireless sensor networks with anycast. **Networking, IEEE/ACM Transactions on**, v. 18, n. 2, p. 515–528, April 2010. ISSN 1063-6692.

KOCAKULAK, M.; BUTUN, I. An overview of wireless sensor networks towards internet of things. In: **2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)**. [S.l.: s.n.], 2017. p. 1–6.

LESSMANN, J.; HEIMFARTH, T.; JANACIK, P. Shox: An easy to use simulation platform for wireless networks. In: **Tenth International Conference on Computer Modeling and Simulation (uksim 2008)**. [S.l.: s.n.], 2008. p. 410–415.

LI, J. et al. Connectivity, coverage and placement in wireless sensor networks. **Sensors**, MDPI AG, v. 9, n. 10, p. 7664–7693, Sep 2009. ISSN 1424-8220. Disponível em: <<http://dx.doi.org/10.3390/s91007664>>.

LI, J.; LAZAROU, G. Y. A bit-map-assisted energy-efficient mac scheme for wireless sensor networks. In: **Third International Symposium on Information Processing in Sensor Networks, 2004. IPSN 2004**. [S.l.: s.n.], 2004. p. 55–60.

LIU, S.; FAN, K. W.; SINHA, P. Cmac: An energy efficient mac layer protocol using convergent packet forwarding for wireless sensor networks. In: **2007 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks**. [S.l.: s.n.], 2007. p. 11–20. ISSN 2155-5486.

LU, G.; KRISHNAMACHARI, B.; RAGHAVENDRA, C. S. An adaptive energy-efficient and low-latency mac for data gathering in wireless sensor networks. In: **18th International Parallel and Distributed Processing Symposium, 2004. Proceedings**. [S.l.: s.n.], 2004. p. 224–.

MANJESHWAR, A.; AGRAWAL, D. P. Teen: a routing protocol for enhanced efficiency in wireless sensor networks. In: **Proceedings 15th International Parallel and Distributed Processing Symposium. IPDPS 2001**. [S.l.: s.n.], 2001. p. 2009–2015. ISSN 1530-2075.

MENDES, L. D. P.; RODRIGUES, J. J. Review: A survey on cross-layer solutions for wireless sensor networks. **J. Netw. Comput. Appl.**, Academic Press Ltd., London, UK, UK, v. 34, n. 2, p. 523–534, mar. 2011. ISSN 1084-8045. Disponível em: <<http://dx.doi.org/10.1016/j.jnca.2010.11.009>>.

PATIL, M.; BIRADAR, R. C. A survey on routing protocols in wireless sensor networks. In: **2012 18th IEEE International Conference on Networks (ICON)**. [S.l.: s.n.], 2012. p. 86–91. ISSN 1531-2216.

PHAM, T. H. et al. Design and implementation of a centralized sensor protocol for information via negotiation. In: **2010 Second International Conference on Communication Software and Networks**. [S.l.: s.n.], 2010. p. 88–92.

POLASTRE, J.; HILL, J.; CULLER, D. Versatile low power media access for wireless sensor networks. In: **Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems**. New York, NY, USA: ACM, 2004. (SenSys '04), p. 95–107. ISBN 1-58113-879-2. Disponível em: <<http://doi.acm.org/10.1145/1031495.1031508>>.

SMARAGDAKIS, G.; MATTA, I.; BESTAVROS, A. **SEP: A Stable Election Protocol for clustered heterogeneous wireless sensor networks**. OpenBU, 2004. Disponível em: <<https://open.bu.edu/handle/2144/1548>>.

SRIVASTAVA, V.; MOTANI, M. Cross-layer design: a survey and the road ahead. **IEEE Communications Magazine**, v. 43, n. 12, p. 112–119, Dec 2005. ISSN 0163-6804.

SUMATHI, R.; SRINIVAS, M. A survey of qos based routing protocols for wireless sensor networks. v. 8, p. 589–602, 12 2012.

TAN, H.-X.; CHAN, M. C. A2-mac: An adaptive, anycast mac protocol for wireless sensor networks. In: **WCNC**. [S.l.]: IEEE, 2010. p. 1–6. ISBN 978-1-4244-6398-5.

THUBERT, P. et al. Rpl: Ipv6 routing protocol for low power and lossy networks. **IETF**, RFC 6550, 03 2012.

VERMA, A. et al. Survey of mac protocol for wireless sensor networks. In: **2015 Second International Conference on Advances in Computing and Communication Engineering**. [S.l.: s.n.], 2015. p. 92–97.

VURAN, I. F. A. M. C. **Wireless Sensor Networks (Advanced Texts in Communications and Networking)**. [S.l.]: Wiley, 2010, 2010. ISBN 9780470036013.

VURAN, M. C.; AKYILDIZ, I. F. Error control in wireless sensor networks: A cross layer analysis. **IEEE/ACM Transactions on Networking**, v. 17, n. 4, p. 1186–1199, Aug 2009. ISSN 1063-6692.

XU, Y.; HEIDEMANN, J.; ESTRIN, D. Geography-informed energy conservation for ad hoc routing. In: **Proceedings of the 7th Annual International Conference on Mobile Computing and Networking**. New York, NY, USA: ACM, 2001. (MobiCom '01), p. 70–84. ISBN 1-58113-422-3. Disponível em: <<http://doi.acm.org/10.1145/381677.381685>>.

YE, W.; HEIDEMANN, J.; ESTRIN, D. An energy-efficient mac protocol for wireless sensor networks. In: **Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies**. [S.l.: s.n.], 2002. v. 3, p. 1567–1576 vol.3. ISSN 0743-166X.

YE, W.; HEIDEMANN, J.; ESTRIN, D. Medium access control with coordinated, adaptive sleeping for wireless sensor networks. **ACM/IEEE Transactions on Networking**, v. 12, n. 3, p. 493–506, jun. 2004. A preprint of this paper was available as ISI-TR-2003-567. Disponível em: <<http://www.isi.edu/%7ejohnh/PAPERS/Ye04b.html>>.

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

ZORZI, M.; RAO, R. R. Geographic random forwarding (geraf) for ad hoc and sensor networks: multihop performance. **IEEE Transactions on Mobile Computing**, v. 2, n. 4, p. 337–348, Oct 2003. ISSN 1536-1233.