

**PROJETO INTEGRADO PARA COLETA DE
DADOS EM REDES DE SENSORES SEM FIO**

FELIPE DOMINGOS DA CUNHA

**PROJETO INTEGRADO PARA COLETA DE
DADOS EM REDES DE SENSORES SEM FIO**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR: ANTONIO A. F. LOUREIRO

Belo Horizonte

Março de 2009

© 2009, Felipe Domingos da Cunha.
Todos os direitos reservados.

C972p Cunha, Felipe Domingos da
Projeto Integrado para Coleta de Dados em Redes
de Sensores Sem Fio / Felipe Domingos da Cunha. —
Belo Horizonte, 2009
xxiv, 75 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de
Minas Gerais

Orientador: Antonio A. F. Loureiro

1. Rede de Sensores Sem Fio. 2. Redes Ad-hoc.
3. Projeto Integrado. 4. Camada MAC. I. Título.

CDU 519.6*22



UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

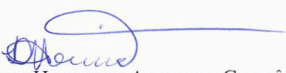
Projeto Integrado para Coleta de Dados em Redes de Sensores Sem Fio

FELIPE DOMINGOS DA CUNHA

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:


PROF. ANTONIO ALFREDO FERREIRA LOUREIRO - Orientador
Departamento de Ciência da Computação - UFMG


PROFA. RAQUEL APARECIDA DE FREITAS MINI - Co-orientadora
Departamento de Ciência da Computação - PUC-MG


PROF. LUIZ HENRIQUE ANDRADE CORRÊA
Departamento de Ciência da Computação - UFLA


PROF. GERALDO ROBSON MATEUS
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 30 de março de 2009.

Dedico esse trabalho aos meus pais, pessoas que me incentivaram e sempre estiveram ao meu lado. A Marcela, minha super irmã terapeuta e Mi, minha irmazinha do coração. Valeu!

AGRADECIMENTOS

Agradeço primeiro a Deus, por ter me dado força, saúde e me abençoado durante essa caminhada. Deus obrigado por não ter me deixado no momento que eu mais precisei.

Agradeço aos meus pais pelo incentivo, ajuda e pelas palavras de apoio em todos os momentos. Sem vocês eu realmente não teria conseguido. Agradeço a Deus pela vida de vocês. Amo vocês!

A minha irmã, amiga e psicóloga Marcela, muito obrigado por tudo. Você que sempre está presente comigo me escutando, dando conselhos. Muito obrigado minha irmã querida!

A minha irmãzinha do peito Michelle, muito obrigado. Você é realmente “sem noção”! Valeu pelo carinho e apoio. Que Deus te recompense tudo àquilo que fez por mim.

Aos meus amigos do laboratório 3053: Daniel, Sumika, Max, Heitor, Guilherme, Leandro, Pedro e DG, valeu galera! Obrigado por todos os momentos de alegria e de muito trabalho que compartilhamos.

Agradeço a Professora Raquel Mini, pela sua ajuda, incentivo, compreensão e pelos conselhos. Realmente você é para mim um exemplo de dedicação e disciplina. Muito obrigado!

Ao Professor Loureiro meus sinceros agradecimentos. Trabalhar com você foi um grande aprendizado pra mim. Obrigado pela oportunidade.

Aos amigos e parentes, quero agradecer pelo carinho e pela força que me deram.

*“Vivendo, se aprende; mas o que se aprende, mais, é só a fazer outras maiores
perguntas.”*

(Guimarães Rosa)

RESUMO

Nas Redes de Sensores Sem Fio os nós possuem severas restrições de energia devido ao fato de possuírem uma bateria pequena e de estarem em locais de difícil acesso para o ser humano. Por isso, essas redes necessitam de protocolos que sejam eficientes no consumo de energia. Diferente das redes tradicionais que são projetadas para acomodar uma grande variedade de aplicações, as RSSFs normalmente são projetadas considerando aplicações bem específicas. Tendo em vista a grande peculiaridade dessas aplicações, podem-se alcançar melhorias significativas de desempenho se o controle e a troca de informações entre duas ou mais camadas forem otimizados de forma que as interações entre as elas sejam mais exploradas. Essa técnica é denominada projeto integrado e tem recebido muita atenção por parte dos pesquisadores da área de RSSFs nos últimos anos. A abordagem integrada de protocolos torna possível uma forte interação entre duas ou mais camadas na pilha de protocolos e possibilita também o compartilhamento de informações em tempo de execução. O objetivo deste trabalho é analisar o desempenho de um projeto integrado de protocolos para viabilizar a adaptação do *duty cycle* em função do tráfego na rede. Assim, propõe-se o algoritmo para a camada de enlace SMAC-DDC (*Sensor-MAC with Dynamic Duty Cycle*) que se beneficia das informações advindas da camada de rede para ajustar a duração do *duty cycle* dos nós sensores com o objetivo de reduzir o consumo de energia e melhorar a coleta dos dados. Resultados de simulações mostram que é possível reduzir a latência e o consumo de energia de uma rede sem comprometer a taxa de entrega de dados ao nó *sink*.

ABSTRACT

In Wireless Sensor Networks, the nodes have severe energy constraints due to the fact of having a small battery and being located in places of difficult access for humans. Therefore, these networks need protocols that are efficient regarding energy consumption. Unlike the traditional networks that are designed to accommodate a wide variety of applications, WSNs are usually designed considering very specific applications. Taking into account the great peculiarity of these applications, you can achieve significant improvements in performance if the control and sharing of information among two or more layers are optimized so that the interactions between them are better exploited. This technique is named cross-layer design and has received much attention from researchers in the area of WSNs in recent years. The cross-layer approach enables a strong interaction between two or more layers in the protocols stack and allows the sharing of information at runtime. The objective of this work is to analyze the performance of a cross-layer design to facilitate the adjustment of the duty cycle depending on the network traffic. Thus, it is proposed the algorithm for the link layer SMAC-DDC (Sensor-MAC with Dynamic Duty Cycle) that receives the information from the network layer to adjust the length of the duty cycle of sensor nodes with the objective of reducing the energy consumption and improve data collection. Results of simulations show that it is possible to reduce the latency and energy consumption of a network without compromising the rate of delivery of data to the sink node.

LISTA DE FIGURAS

1.1	Tipos de comunicação de dados em RSSFs.	2
2.1	Exemplos de nós sensores.	8
2.2	Taxonomia para os protocolos MAC apresentados.	11
2.3	Esquema básico do protocolo T-MAC, com ciclos de atividades adaptáveis [Van Dam & Langendoen, 2003].	13
2.4	Diagrama do funcionamento WiseMAC [Enz et al., 2004].	14
2.5	Períodos de escuta e dormência [Ye et al., 2002].	15
2.6	Formato de um <i>frame</i> no S-MAC [Chen, 2005].	16
2.7	Pontos de verificação em um <i>frame</i> S-MAC [Chen, 2005].	17
2.8	Períodos no S-MAC [Chen, 2005].	19
2.9	Funcionamento do protocolo <i>Directed Diffusion</i> [Intanagonwiwat et al., 2000].	23
2.10	Ilustração de um Projeto Integrado.	25
2.11	Diferentes tipos de comunicação direta entre camadas.	27
2.12	Procedimento de escalonamento do protocolo POWER [Wang & Wang, 2004].	28
2.13	Modificação na arquitetura tradicional de camadas.	29
3.1	Compartilhamento de informação entre camadas.	32
3.2	Etapas de funcionamento do Projeto Integrado.	34
3.3	Pontos de Controle.	36
4.1	Protocolos dos cenários avaliados.	39
4.2	Distância mínima para ocorrência dos eventos fixos.	42
4.3	Latência média variando o <i>duty cycle</i> no protocolo S-MAC.	46
4.4	Mapa do <i>duty cycle</i> dos nós na rede em diferentes momentos.	48
4.5	Comparação da latência para diferentes valores de α	50
4.6	Comparação da latência para diferentes formas de evento.	51
4.7	Comparação do consumo de energia para diferentes valores de α	53

4.8	Comparação do consumo de energia para diferentes formas de evento. . . .	54
4.9	Taxa de entrega de dados para diferentes valores de α	55
4.10	Comparação da taxa de entrega para diferentes formas de evento.	56
A.1	Latência média acumulada para distância mínima igual a 20 e 40 metros. .	64
A.2	Latência média acumulada para distância mínima igual a 60 e 80 metros. .	65
A.3	Energia residual média para distância mínima igual a 20 e 40 metros. . . .	66
A.4	Energia residual média para distância mínima igual a 60 e 80 metros. . . .	67
A.5	Taxa de entrega para distância mínima igual a 20 e 40 metros.	68
A.6	Taxa de entrega para distância mínima igual a 60 e 80 metros.	69

LISTA DE TABELAS

2.1	Comparação das interações nos algoritmos <i>Diffusion</i> [Heidemann & Silva, 2003]. A marcação do asterisco (*) indica que o pacote foi enviado para todos os nós via inundação.	24
3.1	Valores de <i>duty cycle</i> definidos para os protocolos.	33
4.1	Parâmetros de simulação utilizados.	41
4.2	Resultados do Projeto Experimental 2^3r	45
4.3	Latência do primeiro pacote para diferentes valores de α	49
B.1	Resultados do Projeto Experimental 2^3r	74

LISTA DE ALGORITMOS

1	SMAC-DDC – Recebendo um pacote	35
---	--	----

SUMÁRIO

Agradecimentos	ix
Resumo	xiii
Abstract	xv
Lista de Figuras	xvii
Lista de Tabelas	xix
1 Introdução	1
1.1 Motivação	1
1.2 Objetivo e Resultados Obtidos	4
1.3 Organização do Trabalho	4
2 Trabalhos Relacionados	7
2.1 Redes de Sensores Sem Fio	7
2.2 Camada MAC em RSSFs	9
2.2.1 Desafios	9
2.2.2 Protocolos MAC para RSSFs	10
2.3 <i>Sensor MAC</i> (S-MAC)	15
2.3.1 Funcionamento Básico	15
2.3.2 Períodos de Escuta e Dormência	16
2.3.3 Sincronização, Tabela de Programação e Lista de Vizinhos	17
2.3.4 Descoberta de Vizinhos	18
2.4 Camada de Rede em RSSFs	19
2.4.1 Desafios	19
2.4.2 Protocolos de Roteamento para RSSFs	20

2.5	Directed Diffusion	22
2.5.1	Funcionamento Básico	22
2.6	Projeto Integrado de Camadas	25
2.6.1	Introdução	25
2.6.2	Comunicação Direta entre Camadas	26
2.6.3	Modificação na Arquitetura Tradicional	29
2.7	Conclusões	30
3	S-MAC com Ciclo de Trabalho Dinâmico	31
3.1	Integração entre S-MAC e Directed Diffusion	32
3.2	Adaptação do <i>Duty Cycle</i>	33
3.3	Pontos de Controle	35
3.4	Conclusão	36
4	Resultados de Simulação	39
4.1	Cenário de Simulação	40
4.2	Projeto Experimental	42
4.3	Funcionamento Básico	47
4.4	Latência	47
4.5	Consumo de Energia	52
4.6	Taxa de Entrega	53
4.7	Conclusão	55
5	Conclusões e Direções Futuras	57
	Referências Bibliográficas	59
A	Variação da Distância Mínima dos Eventos Estáticos	63
A.1	Latência	63
A.2	Consumo de Energia	66
A.3	Taxa de Entrega	68
B	Projeto Experimental Fatorial	71
B.1	Terminologia	71
B.2	Projeto Experimental 2^k r	72

INTRODUÇÃO

1.1 Motivação

Rede de Sensores Sem Fio (RSSFs) é um tipo especial de redes *ad-hoc*, cujos nós não precisam de uma infraestrutura para se comunicarem [Akyildiz et al., 2002]. Essa rede é composta por milhares de nós sensores autônomos, depositados aleatoriamente em uma determinada área, para coletar dados do ambiente e transmiti-los ao usuário final. Esses nós sensores são equipados com um processador, memória, rádio, sensor e bateria. Entretanto, para diminuição de custos e facilitar a deposição, esses nós possuem seu tamanho reduzido, o que reduz também a capacidade de seus componentes como limitações no processamento, baixa capacidade de armazenamento e baterias com duração limitada.

Devido a essas características, as RSSFs apresentam vários desafios, sendo a restrição de energia o maior deles. Essas redes são freqüentemente projetadas para monitorar ambientes de difícil acesso ao ser humano. Logo, a troca das baterias dos nós sensores se torna inviável, o que exige que essas redes possuam mecanismos para controlar o consumo de energia e que supere falhas na comunicação, contornando problemas como, por exemplo, a perda de nós sensores por falta de energia ou por ação de algum agente externo.

Um dos principais objetivos nas RSSFs é transportar as informações sensorizadas pelos nós sensores até o nó *sink*, responsável pela comunicação entre a rede de sensores e outras redes. Assim, a comunicação de dados é uma tarefa de grande importância e pode ser classificada em três tipos como mostrado na figura 1.1. A primeira ocorre dos sensores para o nó *sink* e tem como objetivo a coleta de dados, como ilustrado na figura 1.1-(a). Na figura 1.1-(b), a comunicação ocorre entre os nós sensores e tem como objetivo a cooperação de dados e, na figura 1.1-(c), a comunicação ocorre do nó

sink para os nós sensores com o objetivo de fazer disseminação de dados, informações e requisições para os nós da rede. Na coleta de dados, a comunicação é usada para enviar informações sensoriadas e coletadas pelos nós ou para enviar uma informação de controle para o nó *sink*. Na cooperação de dados, a comunicação é utilizada para realizar alguma colaboração. Um exemplo é quando os sensores necessitam trocar informações sobre a posição de um objeto sensoriado e enviá-la para o nó *sink*. Na disseminação de dados, a comunicação se inicia no nó *sink* propagando informações para um determinado grupo de nós da rede [Machado et al., 2005].

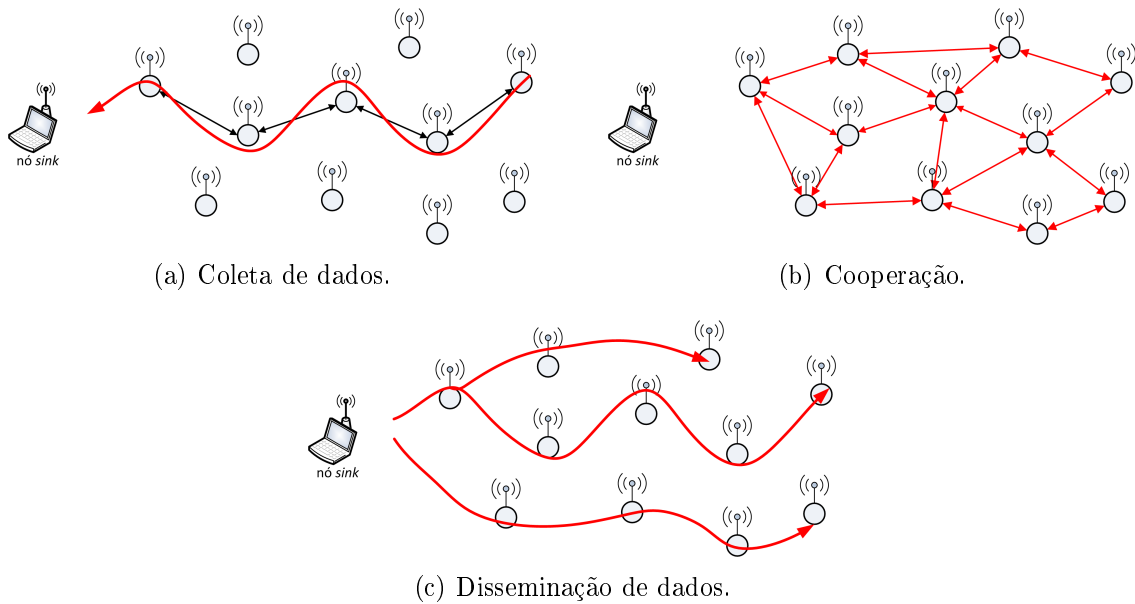


Figura 1.1. Tipos de comunicação de dados em RSSFs.

Comunicação é uma das fontes de maior consumo de energia em uma RSSF. Segundo Pottie & Kaiser [2000], a energia dissipada para a realização de uma transmissão está diretamente relacionada à distância que a mensagem será enviada. Assim, surge a necessidade de projetar protocolos cientes do consumo de energia e que gerenciem o funcionamento do rádio desligando-o em períodos de inatividade. Além disso, esses protocolos devem adotar técnicas para reduzir as transmissões desnecessárias, minimizar o *overhead* na rede e também utilizar uma comunicação com vários saltos.

As camadas responsáveis pela comunicação em RSSFs são as camadas de enlace e de rede. A camada de enlace tem como principal função controlar o acesso ao meio para realizar transmissões em um determinado momento. Esse controle deve evitar que dois ou mais nós transmitam ao mesmo tempo, diminuindo a quantidade de colisões e retransmissões de quadros. Além disso, essa camada também deve controlar os modos de operação do rádio do nó sensor, pois, em momentos de ociosidade na rede, o rádio

deve ser desligado para reduzir o consumo de energia. Outra funcionalidade implementada nessa camada é o controle de erros na transmissão de quadros, aumentando assim a integridade dos dados recebidos. Todas essas tarefas devem ser executadas tendo em vista a conservação da energia.

A função da camada de rede na comunicação de dados é definir uma rota entre a origem e o destino da mensagem [Tanenbaum, 2003]. Neste tipo de rede, a comunicação entre dois nós distantes é feita com a utilização de nós intermediários (comunicação com múltiplos saltos), com o objetivo de reduzir o consumo de energia. Caso alguma rota seja perdida, a camada de rede deverá o caminho entre a origem e o destino da mensagem. Outra funcionalidade que essa camada deverá prover é um esquema de endereçamento para os nós na rede com a finalidade de diferenciá-los. Por se tratar de um tipo de rede com características bem peculiares essas redes requerem que novas técnicas de endereçamento dos nós sejam propostas.

Normalmente, os protocolos tradicionais existentes na literatura para essas camadas não lidam com as especificidades encontradas nesse tipo de rede. Assim, novos protocolos devem ser projetados considerando as características dessas redes, como também os requisitos da aplicação de monitoração [Puccinelli & Haenggi, 2005]. Entretanto, para que o projeto de um novo protocolo considere essas características e requisitos é necessário que os demais protocolos na rede trabalhem de forma cooperativa, trocando informações entre si, com o objetivo de maximizar o desempenho da rede. Para isso, surge o Projeto Integrado de Camadas, um paradigma para a concepção de protocolos que explorem a interação entre os protocolos das camadas adjacentes ou não [Kawadia & Kumar, 2005].

A principal característica do projeto integrado de camadas é romper a estrutura fixa de camadas e permitir a troca de informações entre elas. Com isso, é possível alcançar um melhor desempenho dos protocolos e, conseqüentemente da rede. Essa troca de informações viabiliza também o uso melhor de alguns recursos na rede. Assim, com o projeto integrado, é possível alcançar um melhor gerenciamento desses recursos em todas as camadas [Goldsmith & Wicker, 2002].

Dentre os recursos que devem ser gerenciados em todas as camadas destaca-se a energia. Esse é um recurso escasso e que deve receber grande atenção por todos os protocolos na pilha. Entretanto, uma das maneiras encontradas pela camada de enlace para reduzir o consumo de energia é fracionar o tempo em que o rádio do nó sensor permanece ligado. Dessa maneira, o funcionamento do rádio é alternado em períodos de escuta e dormência. Nos períodos de escuta o rádio permanece ligado, transmitindo e recebendo dados e nos períodos de dormência o rádio do nó é desligado. A duração desses períodos é determinada por uma fração numérica, denominada ciclo de trabalho

(*duty cycle*).

Neste trabalho, é proposto um projeto integrado entre as camadas de enlace e de rede, com a finalidade de melhorar a comunicação de dados entre os nós sensores. Assim, é permitido que a camada de enlace adapte o modo de operação do rádio do nó sensor em função de como as rotas são estabelecidas pela camada de rede.

1.2 Objetivo e Resultados Obtidos

O objetivo deste trabalho é adaptar o *duty cycle* do nó sensor através do compartilhamento de informações entre as camadas de enlace e rede. Para isso, é proposto um estudo de caso, no qual foram escolhidos dois protocolos: o *Sensor-MAC* [Ye et al., 2004] na camada de enlace e o protocolo *Directed Diffusion* [Intanagonwiwat et al., 2000] na camada de rede. Assim, será proposto um novo protocolo MAC que, pelas informações compartilhadas com o protocolo da camada de rede, ajusta o *duty cycle* do nó. Esse protocolo é denominado de SMAC-DDC (*Sensor-MAC with Dynamic Duty Cycle*). O objetivo principal desse novo protocolo MAC é prolongar o tempo de vida da rede, reduzindo o consumo de energia e a latência na transmissão dos dados sem alterações na quantidade de pacotes entregues ao nó *sink*.

Resultados de simulação mostraram que com a adaptação do *duty cycle* dos nós sensores de acordo com o roteamento, foi possível diminuir a latência dos pacotes que chegaram ao nó *sink* reduzindo também o consumo de energia, sem comprometer a entrega de dados para os cenários avaliados.

1.3 Organização do Trabalho

O presente trabalho encontra-se dividido da seguinte forma: o capítulo 2 apresenta alguns protocolos propostos na literatura para a camada de enlace e camada de roteamento. Também são apresentados alguns trabalhos que já aplicaram o paradigma de projeto integrado de camadas no desenvolvimento de melhorias para protocolos de RSSFs. O capítulo 3 apresenta, de uma forma detalhada, como foi efetuado a adaptação do *duty cycle* para o S-MAC, através do compartilhamento de informação entre as camadas de enlace e rede. O capítulo 4 apresenta os resultados de um projeto experimental que avalia os fatores que influenciam a latência e também os resultados comparativos das simulações realizadas de uma rede com e sem a utilização da abordagem proposta. O capítulo 5 apresenta as conclusões e os trabalhos futuros. No apêndice A, são apresentados resultados de simulação variando a distância mínima dos eventos na rede e

finalmente, no apêndice B é apresentado um resumo de como um Projeto Experimental Fatorial é calculado.

TRABALHOS RELACIONADOS

Este trabalho propõe o *Sensor MAC with Dynamic Duty Cycle* (SMAC-DDC), um protocolo que utiliza a técnica de projeto integrado entre a camada de acesso ao meio e a camada de rede para adaptar o tamanho do *duty cycle* do nó sensor. Esse projeto integra o protocolo *Sensor-MAC* (S-MAC) [Ye et al., 2004] e o protocolo de roteamento *Directed Diffusion* [Intanagonwiwat et al., 2000]. Neste capítulo serão descritos alguns protocolos de controle de acesso ao meio para RSSFs, bem como trabalhos propostos na literatura sobre roteamento nesse tipo de redes. Serão também discutidas as diferentes maneiras de criar um projeto integrado entre camadas.

A seção 2.1 apresenta uma breve introdução sobre as redes de sensores sem fio. A seção 2.2 descreve as características da camada MAC para RSSFs e alguns protocolos MAC propostos na literatura bem como os desafios para desenvolvimento de protocolos de controle de acesso ao meio para RSSFs. Na seção 2.3, o protocolo *Sensor MAC* é apresentado detalhadamente. Na seção 2.4, é discutido os desafios encontrados no projeto de um protocolo de roteamento para RSSFs e apresentados os principais protocolos. A seção 2.5 descreve o protocolo *Directed Diffusion*. Na seção 2.6 são apresentados os desafios e propostas de projeto integrado em RSSFs, e na seção 2.7 as conclusões são apresentadas.

2.1 Redes de Sensores Sem Fio

As RSSFs são um tipo especial de rede *ad hoc* e se diferem das redes tradicionais em muitos aspectos. Por exemplo, possuem centenas de nós com severas restrições de energia e são projetadas para monitorar ambientes onde não é desejável a troca de bateria, devendo possuir mecanismos de autoconfiguração da rede para superar as falhas na comunicação e a perda de nós sensores [Akyildiz et al., 2002]. Os nós

sensores são dispositivos autônomos com capacidade de sensoriamento, processamento e comunicação. Esses nós coletam os dados através do dispositivo de sensoriamento, processam localmente ou em cooperação com os seus vizinhos e transmitem para uma estação de armazenamento (*nó sink*). A figura 2.1 ilustra alguns tipos de nós sensores para RSSF. A interconexão entre eles através de enlaces de comunicação sem fio, com objetivo de realizar qualquer tarefa de sensoriamento, certamente, revolucionará os processos de obtenção e processamento de informações.

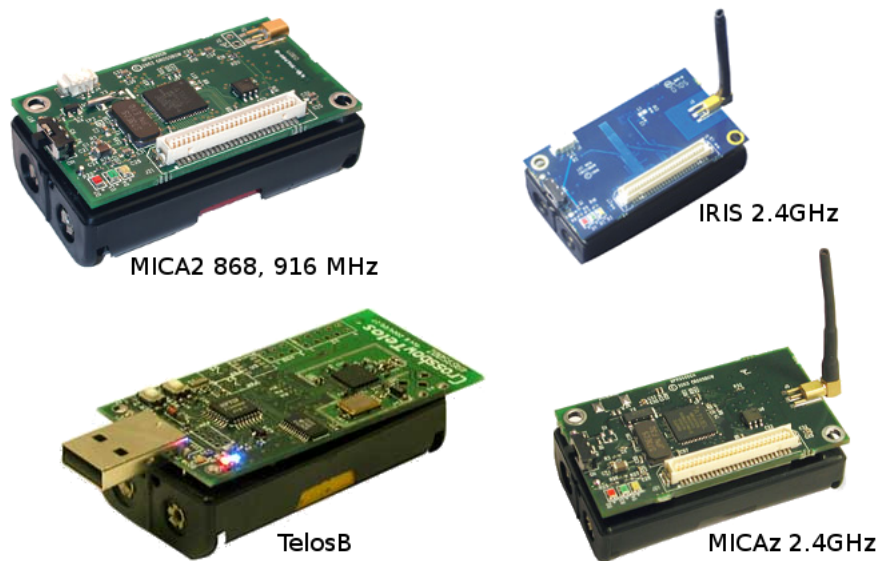


Figura 2.1. Exemplos de nós sensores.

As RSSFs podem ser aplicadas no monitoramento, rastreamento, coordenação e processamento de diversas aplicações. Por exemplo, sensores podem ser interconectados para monitorar e controlar condições ambientais como em florestas, oceanos e planetas. Aplicam-se essas também na prevenção e no auxílio às vítimas de catástrofes naturais tais como terremotos, vulcões, tsunamis, furacões. Na área médica utilizam-se as RSSFs no monitoramento de órgãos vitais, na detecção de substâncias que indicam a existência ou o surgimento de um problema biológico. Utilizam-se as RSSFs na área urbana para melhorar as condições de tráfego e de segurança e na área militar essas redes são utilizadas para detectar movimentos de inimigos, explosões e presença de materiais perigosos.

Atualmente as RSSFs despertam um grande interesse dos pesquisadores, devido aos problemas que surgem como consequência dos requisitos de aplicações e das restrições existentes. Dentre as restrições existentes, destacam-se as baixas taxas de processamento, pouca memória e energia limitada. Em RSSFs, os sensores possuem tamanho

reduzido e tendem a ter um baixo custo de produção, o que leva a utilização de baterias de menor capacidade. Assim, qualquer projeto de melhoria nos protocolos existentes objetivando a economia da energia torna-se de grande.

Outro ponto a ser considerado no projeto de uma RSSF são os requisitos da aplicação que a rede irá sensoriar. Tendo em vista que esse tipo de rede tipicamente é de propósito específico, a escolha de protocolos e hardware deve considerar o tipo de evento a ser sensoriado, a periodicidade desses eventos, a existência de possíveis obstáculos que poderão interferir nas transmissões e também nas alterações da topologia que ocorrerão. Entretanto, os protocolos existentes para redes tradicionais não consideram essas características, pois são projetados para ambientes que não têm essas características.

Nesse contexto, surge à necessidade de projetar protocolos eficientes no consumo de energia e que consideram as características específicas das RSSFs. Este trabalho propõe avaliar o ganho alcançado na latência, consumo de energia e taxa de entrega de dados quando se usa um protocolo que se adequa ao sensoriamento dos dados na rede.

2.2 Camada MAC em RSSFs

2.2.1 Desafios

Os principais desafios para o projeto de protocolos MAC para RSSFs é que essas redes diferem de redes sem fio tradicionais em muitos aspectos, tais como: energia, deposição dos nós, densidade e sensoriamento. Por não tratar de vários desses aspectos, os protocolos MAC tradicionais não podem ser utilizados em RSSFs. Redes de sensores sem fio devem ser eficientes em relação ao consumo de energia, uma vez que os nós sensores são operados por bateria e, devido à alta densidade e aos ambientes inóspitos onde estas redes podem ser implementadas, a recarga dessas baterias é praticamente inviável. Por isso, a característica que deve infligir maior atenção ao se projetar tais redes é o consumo de energia, permitindo o prolongamento no tempo de vida da rede. Em muitas aplicações, os nós são depositados de maneira aleatória e devem se auto-organizar para estabelecer rotas de comunicação. Isso requer que os protocolos para essas redes sejam adaptáveis a mudanças e escaláveis em relação ao número de nós. Outro aspecto que difere RSSFs de redes *ad hoc* é que tanto a densidade da rede, quanto o tipo de sensoriamento dependem do tipo da aplicação [Demirkol et al., 2006]. Finalmente, a maior parte do tráfego nessas redes é desencadeada por sensoriamento de eventos externos que podem ocorrer de uma forma inesperada.

Devido ao fato de protocolos MAC serem influenciados por um grande número de

restrições, faz-se necessário o estudo de alguns atributos importantes desses protocolos. Como o rádio é o dispositivo da rede que mais consome energia e a camada MAC controla os modos de operação do mesmo, desligá-lo quando possível é uma boa técnica de economia de energia [Ye et al., 2004].

Outra maneira de reduzir o consumo de energia na camada MAC é eliminar as principais fontes de desperdício de energia durante a comunicação, que são [Demirkol et al., 2006]:

- *Colisão*: protocolos MAC devem definir quando e como um nó pode acessar o meio e enviar seus dados, evitando que quadros possam colidir. Quando uma colisão ocorre, os dados devem ser descartados e retransmitidos, acarretando em um consumo extra de energia;
- *Idle listening*: é a segunda fonte de desperdício de energia e ocorre quando o rádio escuta o canal de transmissão para receber dados e não existe nenhum dado sendo transmitido;
- *Overhearing*: ocorre quando um nó recebe quadros que são destinados a outros nós. O tráfego desnecessário causado pelo *overhearing* pode ser um fator dominante de desperdício de energia quando o tráfego é pesado e a densidade da rede é elevada;
- *Overhead* de quadros de controle: quadros de controle são usados para diversas funções de protocolos tais como: reserva do canal, sincronização, reconhecimento de quadros de dados. Quadros de controle consomem energia, pois não transportam dados úteis e aumentam o tráfego na rede.
- *Overemitting*: o nó envia um quadro e o nó destino não está pronto para recebê-lo.

2.2.2 Protocolos MAC para RSSFs

Como citado anteriormente, protocolos MAC devem evitar colisão para, dentre outros objetivos, economizarem energia. De acordo com os mecanismos utilizados para evitar colisão, tais protocolos podem ser divididos em dois grupos: baseados em programação e baseados em contenção [Ye & Heidemann, 2003]. A figura 2.2 lista alguns protocolos MAC publicados na literatura de acordo com essa classificação.

Entre os protocolos baseados em programação, a técnica TDMA (*Time Division Medium Access*) tem sido o método mais comum utilizado em RSSFs, no qual um único nó sensor utiliza o canal por um *slot* de tempo. TDMA divide o canal em N *slots* de

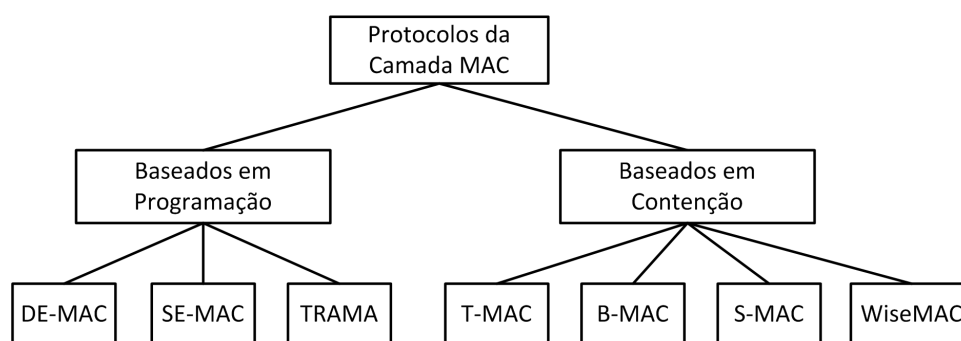


Figura 2.2. Taxonomia para os protocolos MAC apresentados.

tempo, em cada *slot* um nó é autorizado a acessar o meio para transmitir. A maior vantagem de sistemas TDMA é a sua eficiência em energia, uma vez que ele suporta baixo *duty cycle* dos nós, ou seja, baixo ciclo de trabalho ou pequena proporção de tempo durante a qual o nó está ativo. Entretanto, TDMA tem algumas desvantagens que limita seu uso em RSSFs. Uma delas é que esses sistemas normalmente requerem que nós formem agrupamentos e um dos nós dentro do agrupamento deve ser escolhido como o líder. Normalmente os nós são limitados a se comunicarem com o nó líder, ou seja, comunicação entre os nós não é possível. Outra grande desvantagem é sua limitação à escalabilidade e à adaptabilidade em relação ao número de nós, além de ser uma técnica adequada para redes com fluxo contínuo e não para redes baseadas em eventos.

Alguns protocolos para a camada MAC baseados em programação, publicados na literatura são: DE-MAC [Kalidindi et al., 2003], SE-MAC [Lin et al., 2005] e TRAMA [Rajendran et al., 2006]. O DE-MAC (*Distributed Energy Aware MAC*) [Kalidindi et al., 2003] é um protocolo baseado na tecnologia TDMA que utiliza o conceito de períodos de escuta e dormência dos nós. Ele também procura evitar dois grandes problemas com gasto de energia, que são: a escuta ociosa e o *overhearing*. Diferentemente de outros protocolos MAC existentes, o DE-MAC baseia-se no fato que um nó com baixa energia deve ser usado com menor frequência, a fim de realizar balanceamento de carga. O DE-MAC executa um procedimento de eleição local de líder: o nó com mais baixa energia entre todos os nós da rede fica mais tempo em repouso que seus vizinhos. O DE-MAC apresenta algumas vantagens, tais como: não existe perda de quadros devido à colisão, pois dois nós não transmitem no mesmo *slot* de tempo, mas há perdas de quadros devido a outras razões como interferência e esvanecimento do sinal. A vantagem desse protocolo é que nenhum mecanismo de contenção é necessário para um nó começar a transmitir seus quadros, uma vez que os *slots* são pré-definidos para cada nó. Não é necessário nenhum quadro extra de controle (*overhead*) para

contenção.

Outro protocolo para camada MAC baseado em alocação estática do canal é o SE-MAC (*Scalable Energy Efficient Medium Access Control Protocol*) [Lin et al., 2005]. Esse protocolo utiliza a informação local na sincronização dos nós para eliminar colisões. O SE-MAC usa conceitos de períodos de escuta e dormência para evitar *idle listening* e *overhearing*. Para balancear a energia usada na rede, o SE-MAC elege o nó com menor reserva de energia e permite que este permaneça dormindo por mais tempo para economizar energia. Ele também evita naturalmente gasto extra de energia por colisão, uma vez que vários nós podem transmitir dados simultaneamente, sem interferência, em um canal sem fio. No SE-MAC nenhum mecanismo de contenção é necessário para um nó começar a transmitir seus quadros, uma vez que os *slots* são pré-definidos para cada nó e também não é necessário nenhum quadro extra de controle (*overhead*) para contenção.

O protocolo TRAMA (*Traffic Adaptive Multiple Access*) [Rajendran et al., 2006] também é baseado em alocação estática de canal TDMA. Esse protocolo foi desenvolvido para aplicações dirigidas a eventos e tem como objetivo ser eficiente no consumo de energia. Por utilizar este método de alocação do canal, ele garante a não ocorrência de transmissões na transmissão de quadros. Esse protocolo usa um esquema de eleição de líder distribuído baseado na informação acerca do tráfego de cada nó para determinar qual deles irá transmitir em um particular *slot* de tempo. Essas informações são baseadas em escalas anunciadas previamente pelos transmissores, que por sua vez são obtidas pela troca de informações entre a vizinhança a dois saltos de distância. Além disso, esse algoritmo de eleição não faz reserva de *slot* para nós que não têm tráfego de envio.

Diferentemente de protocolos baseados em programação, protocolos baseados em contenção não dividem o canal em sub-canais, nem pré-locam o canal para cada nó. Ao invés disso, o canal é dividido por todos os nós e alocado sob demanda. Um mecanismo de contenção é empregado para decidir qual nó terá direito de acessar o meio em determinado instante. Protocolos baseados em contenção têm algumas vantagens se comparados a protocolos baseados em programação. Eles são mais flexíveis às mudanças na topologia e densidade da rede. Também não necessitam de formação de agrupamento e a comunicação entre nós é permitida e são mais adequados para aplicações orientadas a eventos. Porém, protocolos baseados em contenção também possuem desvantagens, sendo a principal delas sua ineficiência em relação ao uso da energia da rede, isto porque os nós escutam o meio durante todo o tempo [Ye & Heidemann, 2003].

Na literatura são encontrados alguns protocolos baseados em contenção

como: T-MAC [Van Dam & Langendoen, 2003], B-MAC [Polastre et al., 2004], WiseMAC [Enz et al., 2004] e S-MAC [Ye et al., 2002]. O T-MAC (*Time-out-MAC*) [Van Dam & Langendoen, 2003] é um protocolo projetado para aplicações dirigidas a eventos que geram poucos dados, insensíveis à latência e com transmissão contínua ou periódica de dados. A proposta do T-MAC é reduzir o tempo de escuta ociosa transmitindo todas as mensagens em rajadas de tamanhos variáveis e colocar os nós em estado de dormência entre as rajadas. Para manter o tempo de atividade ou a duração da rajada ótima em relação à carga variável, o protocolo determina dinamicamente o tamanho de cada rajada. Para determinar o fim do tempo de atividade, o protocolo simplesmente interrompe esse tempo quando não existe transmissão. O controle do tempo de atividade é feito pelo temporizador T_A , que ao seu término coloca o rádio em repouso, conforme ilustrado na figura 2.3. As mensagens recebidas durante o tempo de repouso são armazenadas e transferidas em rajadas de tamanho variável, no início do próximo tempo de atividade [Correia et al., 2005].

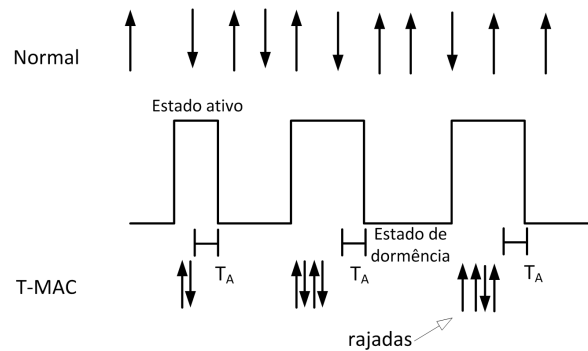


Figura 2.3. Esquema básico do protocolo T-MAC, com ciclos de atividades adaptáveis [Van Dam & Langendoen, 2003].

Em [Polastre et al., 2004], os autores propõem o protocolo B-MAC. A idéia principal é evitar colisões nas RSSFs através da escuta do canal. O método obtém amostras de ruído do meio de transmissão e determina se o canal está ocupado. Para evitar colisão na rede, o B-MAC utiliza a heurística chamada CCA (*Clear Channel Assistent*) que verifica se existe atividade no canal de transmissão. Essa heurística baseia-se na amostragem do nível de ruído no meio de transmissão quando não há tráfego na rede. A partir das amostras, o valor máximo de ruído é determinado e é denominado de ruído base. Quando um nó almeja transmitir, ele faz uma amostragem do ruído do meio e compara com o ruído base. Se o valor da amostra é próximo do ruído base, o nó estima que o meio está livre, caso contrário o meio está ocupado e o nó não pode transmitir. O nó amostra o canal periodicamente para verificar se há transmissão. Para o B-MAC a energia consumida pelo nó se baseia na energia consumida pela recepção, pelo trans-

missão, pela amostragem de sinais no meio e pelo período de repouso [Correia et al., 2005].

Outro protocolo conhecido na literatura é o WiseMAC [Enz et al., 2004], desenvolvido com a finalidade de reduzir o consumo de energia desnecessário nesse tipo de rede. Por ser um protocolo que implementa técnicas de CSMA não persistente com amostragem de preâmbulo, antes de cada transmissão de dados, o nó envia um preâmbulo para o destino informando o início da transmissão (conforme figura 2.4). Enquanto não existe nenhuma troca de dados, os nós escutam o meio por um tempo tw igual a todos, verificando se existe a necessidade de iniciar uma comunicação. Para cada preâmbulo, o WiseMAC determina um tempo de duração dinâmico, o que reduz o tempo de espera para início de uma transmissão e também o consumo de energia. Esse valor é gerado de acordo com as agendas de cada vizinho e assim todas as transmissões de dados são escalonadas. Como artifício de reduzir a taxa de colisão de quadros, o WiseMAC utiliza diferentes valores para os inícios de preâmbulo de escuta.

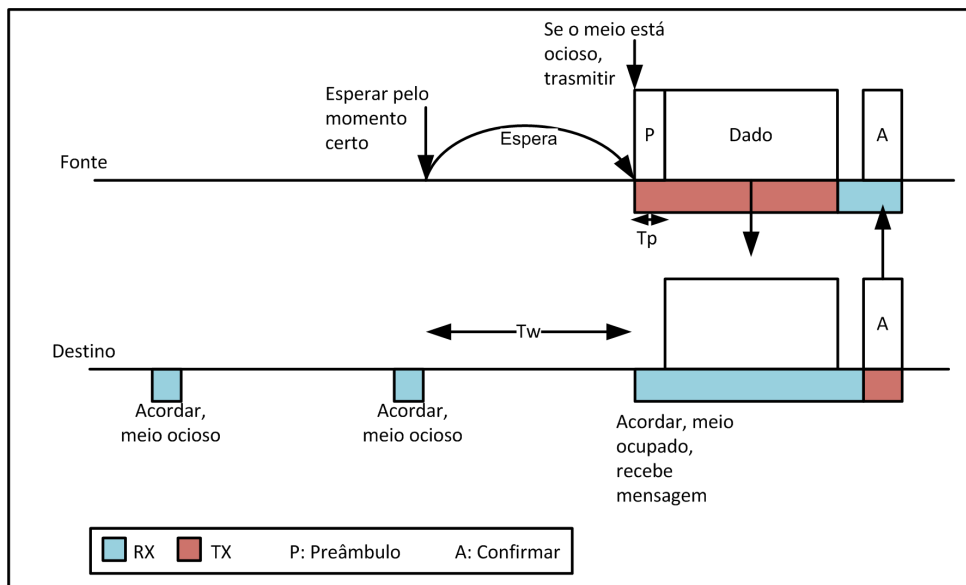


Figura 2.4. Diagrama do funcionamento WiseMAC [Enz et al., 2004].

A próxima seção detalha o protocolo *Sensor MAC* (S-MAC) [Ye et al., 2002], um protocolo baseado em alocação dinâmica de canal que utiliza sincronização para coordenar os modos de operação do rádio. O S-MAC é destinado a aplicações dirigidas a eventos cuja latência não é um fator crucial. O protocolo S-MAC foi utilizado neste trabalho por ser um dos primeiros protocolos MAC proposto especificamente para RSSFs e também por ser o protocolo mais utilizado na literatura concernente. Além disso, esse protocolo trabalha com períodos de *duty cycle* fixos, o que em aplicações de

tráfego variado pode implicar em um mau funcionamento, necessitando, portanto, de adaptações.

2.3 *Sensor MAC* (S-MAC)

Nesta seção é apresentado o protocolo S-MAC [Ye et al., 2002], seu funcionamento básico, suas características e principais técnicas para redução de energia. É apresentado também o funcionamento dos períodos de escuta e dormência no S-MAC, as programações dos nós, sincronização e o funcionamento da lista de vizinhos.

2.3.1 Funcionamento Básico

O protocolo S-MAC apresenta como meta a redução do consumo de energia e a redução de colisões. Para isso, o S-MAC utiliza uma combinação de sincronização de agendas e um esquema de contenção. Para ser eficiente em energia, o S-MAC procura reduzir o desperdício nos principais eventos responsáveis pelo consumo, que são: colisão, *overhearing*, *overhead* e *idle listening*. Para tanto, esse protocolo utiliza três técnicas básicas: introdução de períodos de dormência e escuta; prevenção de colisão e *overhearing* e inserção do conceito de *message passing*.

Em muitas aplicações para RSSFs, os nós podem permanecer ociosos por um longo período, principalmente se nenhum evento é sensoriado, pois a taxa de dados neste intervalo é muito baixa, não sendo necessário que os nós permaneçam escutando o canal durante todo o tempo. O protocolo S-MAC reduz o tempo de escuta do nó e, conseqüentemente, um gasto extra de energia, colocando os nós no modo de dormência. A figura 2.5 ilustra a idéia básica, que é alternar períodos de escuta e dormência do nó. Durante o período de dormência o nó desliga seu rádio e ajusta um temporizador para acordá-lo mais tarde.

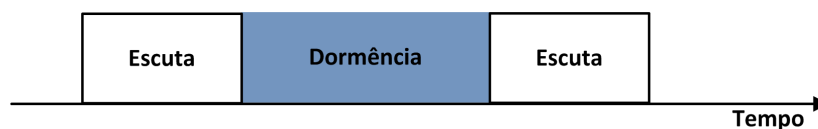


Figura 2.5. Períodos de escuta e dormência [Ye et al., 2002].

No S-MAC, os nós trocam suas agendas de funcionamento com seus vizinhos mais próximos, via difusão. O objetivo é que os vizinhos possam se comunicar mesmo que possuam tabelas diferentes. Nesse protocolo, os nós que possuem a mesma agenda formarão agrupamentos na rede.

Outra característica do protocolo S-MAC é que os nós vizinhos são livres para se comunicarem, não importando sua agenda. Os nós são sincronizam suas agendas com a finalidade de evitar o problema de comunicação dentro dos agrupamentos e interferência. Isto facilita adaptação às mudanças na topologia. A desvantagem é que a latência é incrementada devido ao período de dormência de cada nó.

Nas próximas seções, o protocolo S-MAC será discutido detalhadamente. A seção 2.3.2, explica o funcionamento dos períodos de escuta e dormência. A seção 2.3.3 descreve a sincronização e a tabela de programação e a seção 2.3.4 explica o mecanismo de descoberta de vizinhos.

2.3.2 Períodos de Escuta e Dormência

A principal técnica utilizada pelo protocolo S-MAC para reduzir o consumo de energia é fazer com que cada nó na rede siga um ciclo de escuta e dormência, também conhecido como períodos de *listen* e *sleep*. Um período completo de escuta e dormência é denominado *frame*¹. Durante o período de dormência o nó desliga seu rádio. Desta forma, a energia consumida devido ao *idle listening* pode ser evitada, principalmente quando o tráfego de dados na rede é baixo. Durante períodos de escuta, o nó pode iniciar o recebimento e envio de quadros². O S-MAC fornece um parâmetro ajustável, *duty cycle*, cujo valor representa a razão entre o período de escuta e o tamanho total do *frame* e seu valor pode variar de 1 a 100%. O período de escuta é dividido em duas partes. A primeira, denominada período *SYNC*, é destinada para envio de quadros de sincronização. A segunda, denominada período *DATA*, é designada para transmissão de dados, conforme ilustra a figura 2.6.

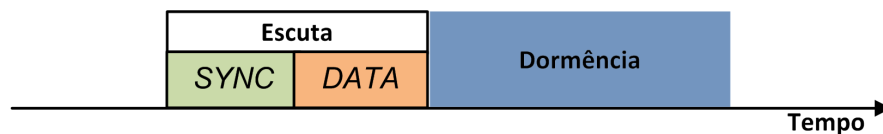


Figura 2.6. Formato de um *frame* no S-MAC [Chen, 2005].

Cada nó no S-MAC deve ter, no mínimo, uma programação, ou seja, um período de escuta e um período de dormência de acordo com o período de dormência e escuta de um de seus vizinhos. Cada programação é controlada por um relógio que pode ser reprogramado quando expirar. Cada *frame* no S-MAC possui três pontos de verificação ou pontos de expiração. Nesses pontos, o S-MAC decide a ação a ser feita no próximo

¹Termo utilizado para indicar um período completo de escuta e dormência.

²Termo utilizado para referenciar a mensagem trocada por entidades da camada MAC.

período de tempo destinado a uma atividade do protocolo. Na figura 2.7, no ponto 2, o S-MAC verifica se existe algum quadro de dados no *buffer* para ser enviado. Em caso afirmativo, o protocolo inicia a detecção do meio para verificar se o mesmo está livre. Em caso negativo, o S-MAC não tentará enviar o quadro de dados neste *frame*. Mesmo que dados sejam recebidos no meio do período *DATA*, o S-MAC não enviará esses dados, mas irá armazená-los no *buffer* até o próximo ponto de verificação 2. O mesmo ocorre para os pontos 1 e 3. No ponto 1, o S-MAC permite que os nós enviem e recebam quadros de sincronização. Se nenhum nó executar esta tarefa e o período *SYNC* terminar, o nó só poderá enviar seu quadro *SYNC* no próximo ponto de verificação 1. No ponto 3, o S-MAC possibilita a dormência ou recepção/envio de dados. Se o nó não executar esta tarefa, ela só poderá ser executada no próximo ponto de verificação 3.



Figura 2.7. Pontos de verificação em um *frame* S-MAC [Chen, 2005].

2.3.3 Sincronização, Tabela de Programação e Lista de Vizinhos

A programação de escuta/dormência exige sincronização entre os nós vizinhos para evitar grandes erros de relógio. Esta atualização normalmente é realizada enviando, periodicamente, um quadro *SYNC*. O período no qual o nó transmite seu quadro *SYNC* é denominado período de sincronização. Esse quadro inclui o endereço do remetente e o tempo para o próximo período de dormência. O próximo período de dormência é relativo ao momento no qual o emissor termina de transmitir o quadro *SYNC*. Os receptores ajustam seus temporizadores imediatamente após o recebimento do quadro *SYNC* e o nó irá dormir assim que o temporizador disparar.

Cada nó no S-MAC possui uma tabela de programação que armazena a programação do nó e a programação de seus vizinhos. Existem dois tipos de tabelas de programação: a programação primária que armazena a programação do próprio nó e a programação secundária que armazena as programações dos seus vizinhos. Todo nó neste protocolo deve possuir uma programação primária. O número de programações que um nó pode ter é limitado, e esse valor pode ser ajustado pelo desenvolvedor da

rede. Espera-se que exista apenas uma programação em toda a rede, para maximizar a economia de energia. Mas a existência de múltiplas programações na rede é mais comum, especialmente em uma rede com vários saltos, na qual os nós não iniciam suas atividades ao mesmo tempo e agrupam-se de maneira aleatória.

No S-MAC, cada nó deve armazenar uma tabela com as informações de todos os seus vizinhos. O número de vizinhos em cada lista é um parâmetro ajustável pelo desenvolvedor da rede e define o número máximo para cada nó. Como na tabela de programação, a lista de vizinhos também é estabelecida durante a troca de quadros *SYNC* entre os nós. Quando um nó recebe uma requisição de envio de dados, primeiramente ele verifica se o nó para o qual ele deve enviar a informação está na sua lista de vizinhos. Em caso negativo, ele recusa a requisição. Se o nó está na em sua lista, ele tentará enviar este quadro.

No S-MAC quando um novo nó é inserido na rede, ele primeiramente escuta por um período fixo, normalmente um período de sincronização. Este período é denominado escuta inicial. Durante este período podem ocorrer dois eventos:

- O nó não recebe nenhum quadro *SYNC* durante o período de escuta inicial. Quando isto ocorre, o nó escolhe, espontaneamente, uma programação e ajusta um relógio para a mesma. A programação escolhida é adicionada à primeira entrada na sua tabela de programações. Para anunciar sua nova programação, o nó transmitirá um quadro *SYNC* no próximo período *SYNC*.
- O nó recebe um quadro *SYNC* antes do fim do período de escuta inicial. Neste caso o nó imediatamente escolhe e segue a programação deste quadro, ao invés de escolher espontaneamente outra no fim do período de escuta inicial. Esta programação é então adicionada à primeira entrada na sua tabela de programações e o nó remetente deste quadro *SYNC* é adicionado à sua lista de vizinhos. O nó precisa anunciar a programação que está seguindo. Ele faz isso transmitindo um quadro *SYNC* no seu próximo período *SYNC*.

2.3.4 Descoberta de Vizinhos

No S-MAC, os nós vizinhos se descobrem trocando quadros de sincronização. A idéia básica da descoberta de vizinhos no S-MAC é fazer com que cada nó execute, periodicamente, a descoberta de vizinhos durante um período de sincronização. Durante o período de descoberta de vizinhos, o nó não dormirá mesmo no seu período de dormência, isto para permitir que o nó escute o meio por um período maior e tenha mais chances de escutar e descobrir um novo vizinho.

A figura 2.8 mostra a relação entre o período de descoberta de vizinhos e outros períodos no S-MAC. Pode-se notar que o período de descoberta de vizinhos pode variar dependendo do número de vizinhos atuais que o nó possui. Este período pode variar de dois períodos de sincronização, se o nó não possui nenhum vizinho conhecido, até 33 períodos de sincronização, dependendo do número de vizinhos conhecidos do nó.

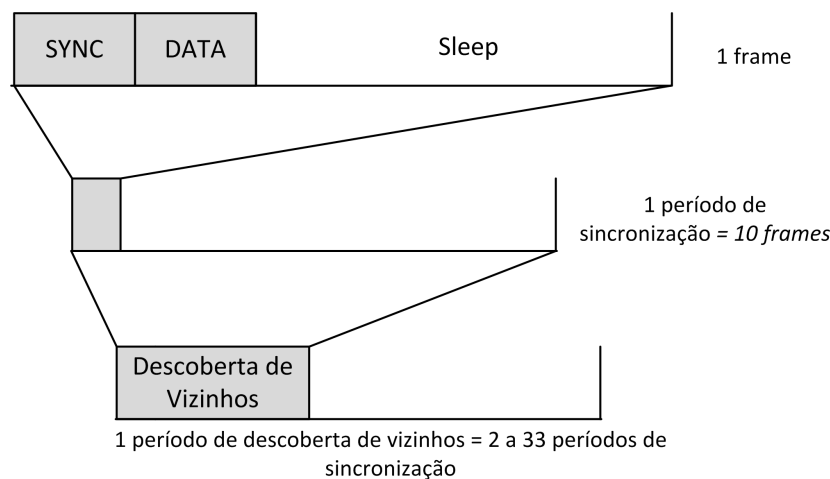


Figura 2.8. Períodos no S-MAC [Chen, 2005].

2.4 Camada de Rede em RSSFs

2.4.1 Desafios

A principal função da camada de rede é prover o serviço de roteamento que pode ser definido como o processo pelo qual a rede consegue identificar o destinatário das mensagens e encontrar um caminho entre a origem e o destino desta mensagem [Ruiz et al., 2004]. A grande meta do roteamento é ser eficiente para entregar dados durante todo o percurso, desde a fonte até o destino do pacote. Em RSSFs a estratégia adotada como roteamento tem um impacto significativo no desempenho, principalmente quando os nós sensores atuam também como roteadores.

Alguns desafios são encontrados em RSSFs para o projeto de protocolos para o roteamento. Nesse tipo de rede os protocolos de roteamento devem considerar a existência de nós não alcançáveis, ou por interferência na transmissão ou perda de conexão ocasionada pela mobilidade dos nós [Jurdak, 2007]. Essa mobilidade introduz um alto grau de complexidade nessas redes, principalmente pelo fato de não existir uma central que controle a posição atual de cada nó, dificultado a criação das rotas e o estabelecimento do roteamento. Outro fator que deve ser tratado pelo roteamento

é a redundância de dados, pois, nesse tipo de rede os dados dos nós fontes em geral referem-se a um mesmo fenômeno aumentando a probabilidade de redundância.

Por fim a energia, um recurso que é escasso nesse tipo de rede. Por isso, o projeto de protocolos de roteamento deve considerar grandemente esse recurso, pois, seu consumo exacerbado implica na morte prematura dos nós deixando a rede desconexa, o que dificulta o roteamento dos pacotes. Na seção a seguir os principais protocolos de roteamento para RSSFs encontrados na literatura serão apresentados.

2.4.2 Protocolos de Roteamento para RSSFs

Técnicas de roteamento são necessárias para enviar os dados entre os nós sensores e o nó *sink*. Muitos protocolos são encontrados na literatura para o roteamento em redes de sensores. Segundo Al-Karaki e Kamal [Al-Karaki & Kamal, 2004], pode-se classificar os protocolos de roteamento encontrados de acordo com a estrutura da rede. Essa taxonomia divide os protocolos da seguinte forma: roteamento plano e roteamento hierárquico.

No roteamento plano todos os nós possuem a mesma função e colaboram entre si para o sensoriamento dos eventos. O evento é sensoriado e os dados são enviados ao nó *sink* utilizando comunicação do tipo múltiplos saltos. Nesse contexto, diversos protocolos são propostos. Dentre eles, encontra-se o protocolo SPIN (*Sensor Protocols for Information via Negotiation*) [Kulik et al., 2002], que usa informações sobre a quantidade de energia disponível em cada sensor para fazer o roteamento. Ele utiliza protocolos de negociação para disseminar as informações de um nó sensor para todos os nós sensores da rede. No SPIN quando um nó percebe que sua energia está perto de um limite pré-estabelecido ele se adapta participando menos da disseminação de dados. Ele também permite ao usuário que faça uma pesquisa a qualquer nó, e esse nó retorna com a resposta imediatamente.

Nessa categoria encontra-se também o protocolo SAR (*Sequential Assignment Routing*) [Sohrabi et al., 2000], que visa facilitar o roteamento com múltiplos saltos. O objetivo desse protocolo é minimizar a média ponderada de métricas de qualidade de serviço (QoS - *Quality of Service*) através do tempo de vida da rede. A decisão de qual rota utilizar leva em consideração três fatores: energia residual, as métricas de QoS de cada caminho e o nível de prioridade dos pacotes.

No roteamento hierárquico, os nós terão papéis diferentes na rede. A definição desses papéis pode ser feita no projeto da rede com a escolha de nós com capacidade diferenciada ou então, durante a execução do algoritmo de roteamento. A rede é dividida em agrupamentos e cada agrupamento terá um nó especial que enviará as

mensagens de seu grupo para o nó *sink*.

Em [Heinzelman et al., 2000, 2002], é proposto o LEACH (*Low Energy Adaptive Clustering Hierarchy*), um algoritmo de roteamento hierárquico, auto-organizável e adaptativo, desenvolvido para RSSFs homogêneas. Nesse algoritmo, os nós se organizam em grupos com um nó agindo como líder do grupo. Os nós enviam os seus dados para o líder e ele se encarrega de enviar todos os dados para o *sink*. Para efetuar este envio, o líder tem um gasto de energia consideravelmente maior. Por isso, o LEACH faz um rodízio aleatório de líderes com o objetivo de não esgotar a energia em um único nó. Assim evita o aparecimento de áreas descobertas na rede. Ele divide a operação da rede em rodadas, onde cada rodada começa organizando os agrupamentos e depois passa para uma fase constante de transmissão de dados.

Durante o período de organização do agrupamento, um nó se elege líder com certa probabilidade, sendo pré-determinado o número de nós P , que serão os possíveis líderes da rede. A decisão de se tornar um líder ou não é feita pela escolha de um número aleatório entre 0 a 1. Caso este número escolhido seja menor que o limite $T(n)$, esse nó será um líder. Este limite $T(n)$ é dado pela expressão:

$$T(n) = \begin{cases} \frac{P}{1 - P^{*(r \bmod \frac{1}{P})}} & \text{se } n \text{ pertence a } G \\ 0 & \text{caso contrário} \end{cases}$$

onde P é a porcentagem desejada de nós líderes na rede, r é a rodada corrente, G é o conjunto de nós que não foram líderes nas últimas $1/P$ rodadas, sendo que a chance de um nó que não foi líder se tornar um vai aumentando a cada rodada que se passa.

Outro protocolo encontrado na literatura é o PEGASIS (*Power-Efficient Gathering in Sensor Information Systems*) [Lindsey & Raghavendra, 2002]. A idéia chave do protocolo PEGASIS é formar uma corrente entre os nós sensores onde cada nó receberá e transmitirá para o vizinho mais próximo. A construção dessa construção inicia do nó mais afastado do nó *sink*, que adiciona o seu vizinho mais próximo à corrente até que todos os nós o façam. Como no algoritmo guloso, as distâncias vizinhas aumentarão gradualmente desde que os nós que estão na corrente não podem ser visitados. Alternativamente, o nó *sink* pode computar esta corrente e transmiti-la a todos os nós sensores.

Para recolher dados em cada rodada cada nó recebe dados de um vizinho, funde-os com seus próprios dados, e transmite-os ao outro vizinho na corrente. Em cada rodada, usa-se um sistema de passagem simbólica do *token* iniciado pelo líder, para começar a transmissão de dados nas extremidades da corrente. Os nós vão passando o *token*

juntamente com os dados até alcançar o líder. A principal vantagem é a diminuição do gasto de energia na rede, utilizando apenas um nó líder em toda rodada. O líder em cada rodada estará em uma posição aleatória na corrente, o que é importante para que os nós morram em posições diferentes. Isso torna a rede robusta às falhas.

Na seção seguinte é apresentado um algoritmo de roteamento centrado em dados: *Directed Diffusion*. Esse protocolo baseia o roteamento de acordo com a disseminação de interesses e também através do estabelecimento das rotas utilizando a definição de gradientes.

2.5 Directed Diffusion

O protocolo *Directed Diffusion* [Intanagonwiwat et al., 2000] tem como objetivo a criação de canais de comunicação eficientes entre os nós sensores e o nó *sink*. Esse protocolo utiliza dois conceitos: roteamento baseado em dados e agregação de dados. O roteamento baseado nos dados ocorre através da requisição de informação pelo nó *sink* chamada de disseminação de interesses. O nó que possui a informação requisitada responderá ao nó *sink*. A agregação de dados ocorre quando os nós intermediários agregam seus dados em um simples pacote com a finalidade de reduzir as transmissões e o volume de dados trafegados na rede. Na seção a seguir, cada fase do protocolo será explicada detalhadamente.

2.5.1 Funcionamento Básico

No protocolo *Directed Diffusion* os pacotes são roteados em função dos atributos ao invés do endereço dos nós. Por isso, nesse protocolo, as tarefas a serem sensoriadas são definidas por um par de atributos e valores e, a descrição de uma tarefa, denominada por *interesse* é disseminado na rede pelo nó *sink* (figura 2.9-(a)).

Para cada tarefa ativa, o nó *sink* irá enviar periodicamente um interesse para todos os seus vizinhos, até alcançar toda a rede. Ao receber um interesse, um nó deve enviar interesses para os seus vizinhos de forma a possibilitar a propagação dos interesses através da rede, inundando toda a rede com pacotes de interesses. Quando um nó recebe um interesse, ele verifica se o mesmo existe na sua tabela. Caso ainda não exista, uma nova entrada na tabela é criada com um gradiente para o nó que enviou o interesse. Caso o interesse recebido esteja presente na tabela de interesses, verifica-se então se existe um gradiente para o nó de origem. Se o gradiente não existir um novo gradiente é criado (figura 2.9-(b)).

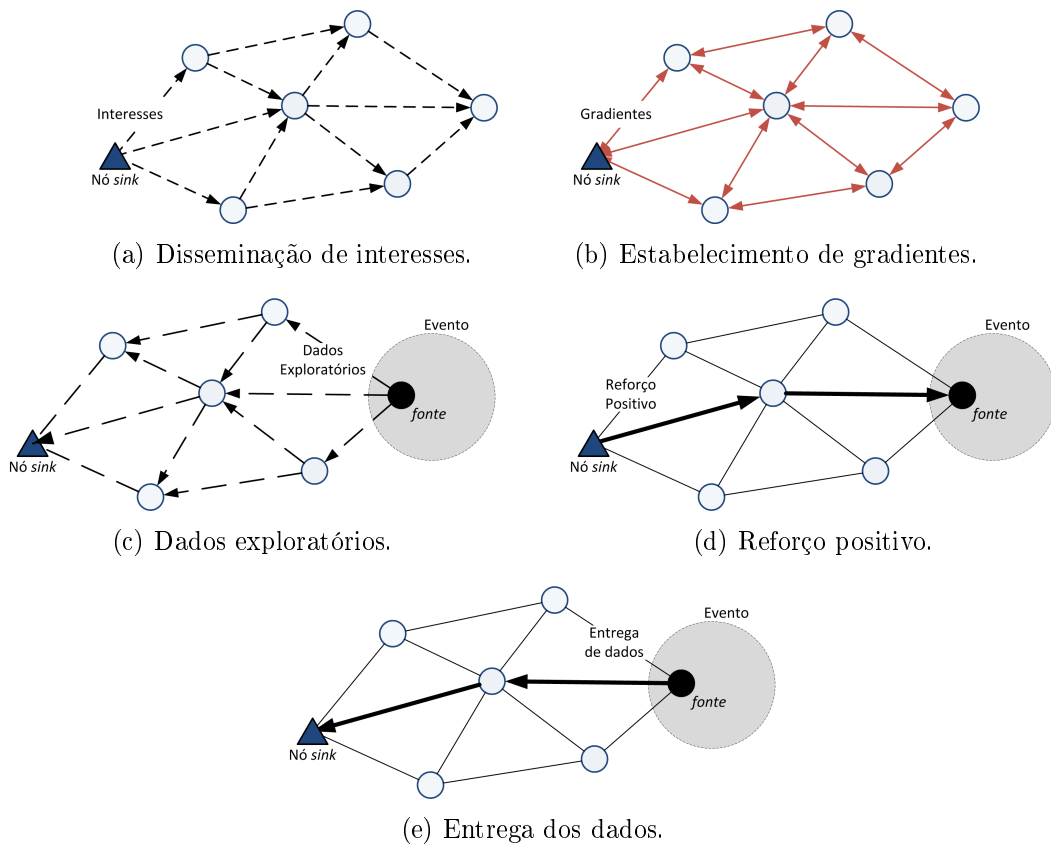


Figura 2.9. Funcionamento do protocolo *Directed Diffusion* [Intanagonwiwat et al., 2000].

Após o recebimento dos interesses e configuração apropriada dos gradientes, o nó liga os seus sensores para monitorar os eventos que foram solicitados a observar. Assim, ao detectar algum evento, o nó processa o dado sensoriado e procura na sua tabela de interesses se existe algum interesse que coincida com o tipo de evento que foi observado. Caso exista algum interesse, o nó inicia o processo de envio de dados exploratórios (figura 2.9-(c)).

O envio de dados exploratórios é feito pelo nó sensor que detectou o evento. Esse nó enviará pacotes do tipo dado exploratório por meio de uma comunicação difusão até alcançar o nó *sink*. Esse tipo de pacote informará ao nó *sink* que a coleta de dados acerca de um evento iniciará. Em geral, esses dados são enviados pelo nó fonte a uma baixa taxa de transmissão.

O primeiro passo a ser realizado pelo nó *sink* ao receber pacotes de dados exploratórios é enviar um pacote do tipo reforço positivo em direção oposta ao pacote recebido (figura 2.9-(d)). Esse reforço indicará para o nó fonte que ele poderá enviar os dados coletados pela rota que foi reforçada a uma taxa maior (figura 2.9-(e)). O

reforço positivo pode ser utilizado também para o reparo de caminhos que falharam. Quando um determinado nó detecta que um caminho falhou, o nó realiza um processo de reforço positivo de alguns de seus vizinhos para restabelecimento da comunicação e evitar a interrupção na entrega de dados. Dessa forma, novos caminhos podem ser utilizados de forma a tentar garantir a taxa de eventos solicitada no interesse.

Outro tipo de reforço utilizado neste protocolo é o reforço negativo. O objetivo desse reforço é amenizar o problema do uso de múltiplos caminhos que acabam consumindo mais energia da rede. Quando um novo caminho é reforçado, deve-se realizar um reforço negativo no caminho utilizado anteriormente. Esse reforço negativo é feito através do envio de um pacote especial do tipo reforço negativo aos vizinhos que não são mais úteis.

Como variações do protocolo original, o protocolo *Directed Diffusion* apresenta mais duas versões. A versão original denominada *Two-phase pull diffusion* constitui-se de cinco etapas, conforme apresentado. As demais versões são apenas extensões da versão inicial com a exclusão de algumas etapas. A versão *Push diffusion* [Heidemann & Silva, 2003] é uma versão reativa na qual as rotas e os gradientes são configurados apenas no momento da detecção de um evento. Essa versão é indicada para redes nas quais os nós sensores trabalham com baixo *duty cycle* na maior parte do tempo. E a segunda versão, *One-phase pull diffusion* [Heidemann & Silva, 2003], é uma versão proativa do protocolo que minimiza a quantidade de inundações feitas pelos pacotes de interesses. A tabela 2.1 apresenta a sequência de dados trafegados na rede de acordo com as fases contidas em cada versão.

Protocolo	Nó sink	Nó fonte
<i>Two-phase pull</i>	Interesses* Reforço positivo (resposta ao dado exploratório)	Dados exploratórios* Dados
<i>One-phase pull</i>	Interesses*	Dados
<i>Push</i>	Reforço positivo (resposta ao dado exploratório)	Dados exploratórios* Dados

Tabela 2.1. Comparação das interações nos algoritmos *Diffusion* [Heidemann & Silva, 2003]. A marcação do asterisco (*) indica que o pacote foi enviado para todos os nós via inundação.

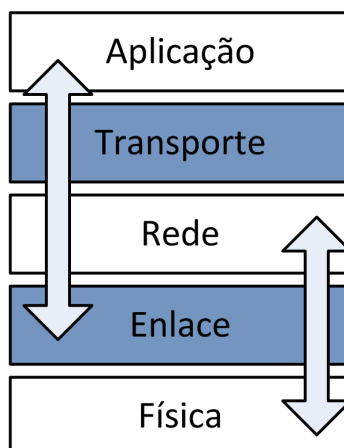


Figura 2.10. Ilustração de um Projeto Integrado.

2.6 Projeto Integrado de Camadas

Nesta seção será apresentado o conceito de projeto integrado entre camadas e, a partir de uma taxonomia proposta na literatura, os trabalhos encontrados que fazem uso desse conceito são classificados.

2.6.1 Introdução

A comunicação baseada na arquitetura de rede em camadas tipicamente separa as tarefas por nível, com a clara definição da funcionalidade e serviço que cada camada deverá prover. A interação entre essas camadas na pilha de protocolos ocorre somente através de serviços pré-definidos disponíveis apenas entre camadas que são vizinhas. Essa interação é restrita e não permite nenhum outro fluxo de informação entre camadas não adjacentes, o que dificulta o monitoramento de determinados recursos em alguns tipos de rede.

Em contraste, outro tipo de abordagem para a interação entre camadas existente na literatura é o conceito de projeto integrado de camadas, que atenta para a rica exploração da troca de informação entre camadas adjacentes ou não, conforme ilustra a figura 2.10, com o objetivo de proporcionar ganhos no desempenho geral da rede [Jurda, 2007]. Srivastava & Motani [2005] propuseram uma definição para projeto integrado como: “projeto de protocolos pela violação da tradicional arquitetura de comunicação em camadas”. Seguindo esse paradigma, novos protocolos foram projetados com o objetivo de melhorar o seu funcionamento e propiciar o gerenciamento de alguns recursos em mais de uma camada ao mesmo tempo.

Em [Srivastava & Motani, 2005], é também proposto uma classificação dos proto-

colos que utilizam a arquitetura integrada de camadas de duas maneiras: na primeira abordagem separam-se os protocolos pela maneira que a violação da arquitetura de camadas ocorre, podendo acontecer de quatro maneiras: criação de novas interfaces; união de camadas adjacentes; projeto em conjunto de protocolos e calibração vertical de parâmetros. Em outra vertente, dividem-se os protocolos de acordo com sua implementação. Nesta abordagem três categorias são instituídas: comunicação direta entre as camadas, base de dados compartilhada através das camadas e a criação de novas abstrações. Esta é uma classificação proposta de forma genérica e aplicável para qualquer tipo de rede sem fio.

Neste trabalho optou-se por dividir os protocolos encontrados de duas maneiras: agrupando os trabalhos que efetuam apenas uma comunicação direta entre as camadas adjacentes ou não por intermédio de uma interface. Depois são agrupados os trabalhos que propõem algum tipo de modificação na arquitetura tradicional de camadas por adição de módulos externos. Nas seções seguintes, as principais características de cada classe serão detalhadas e também os trabalhos encontrados na literatura serão divididos e atribuídos à sua classe.

2.6.2 Comunicação Direta entre Camadas

Uma maneira simples de permitir o compartilhamento de informações entre as camadas é habilitar a comunicação entre elas. Para isso pode ser usado uma interface entre as camadas que intermedeia a troca de informações entre elas. Essa comunicação é possibilitada através da adição de informações no cabeçalho do pacote de forma que outras camadas envolvidas no projeto integrado tenham acesso à informação.

O protocolo proposto nesse trabalho se encaixa nessa classificação, pois, as informações trocadas entre a camada de rede e de enlace utilizam o artifício de trocar essas informações através de dados inseridos no cabeçalho do pacote. Na figura 2.11 vemos as diferentes maneiras de como pode ocorrer a comunicação direta entre as camadas na pilha de protocolos e a seguir alguns protocolos desenvolvidos para as RSSFs serão descritos.

O trabalho proposto em [Van Hoesel et al., 2004] apresenta a utilização do projeto integrado entre as camadas MAC e de roteamento. Nesse exemplo, foi usado o protocolo EMACs que minimiza o consumo de energia da rede dividindo as operações dos nós entre ativo, passivo e adormecido. Os nós no estado ativo encaminharão as mensagens dos nós no estado passivo, os passivos guardarão informações dos seus vizinhos ativos e os nós no estado adormecido permanecem com baixo consumo de energia por uma determinada quantidade de tempo. Para o roteamento o algoritmo escolhido é o ESR

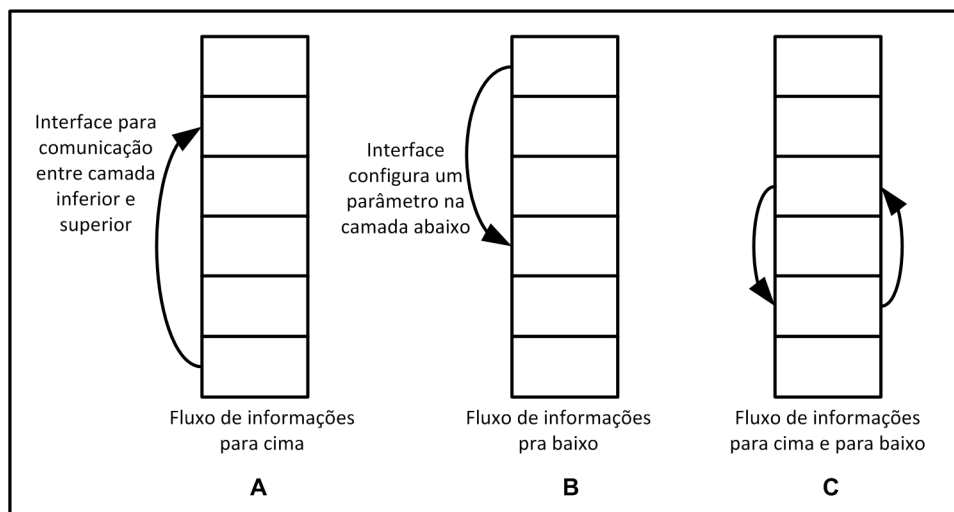


Figura 2.11. Diferentes tipos de comunicação direta entre camadas.

(*EYES Source Routing*), um protocolo sob-demanda, dinâmico, que se auto inicia na rede com o objeto de realizar um roteamento com vários saltos entre a fonte e destino da informação. A integração entre eles ocorre quando o roteamento necessita conhecer a topologia da rede para entregar os pacotes aos seus destinatários e, acessando a tabela de vizinhos ele obtém conhecimento da local topologia e torna o roteamento mais eficiente. Essa informação é obtida pela camada MAC, que possui uma lista dos nós vizinhos ativos aptos a participar do roteamento. Essa abordagem mostrou-se eficiente, principalmente em ambientes nos quais os nós são móveis devido a alternância de papéis entre eles (ativo/passivo).

Em [Fang & McDonald, 2004], é proposto um protocolo para roteamento em redes de sensores que utiliza técnicas básicas de codificação para reduzir a interferência entre duas ou mais rotas. A informação compartilhada entre as camadas, conciliada com a codificação ajuda a definir os caminhos para o roteamento de modo que uma rota não exerce interferência sobre a outra. Para isto cada rota é construída utilizando uma seqüência de códigos que indicam o nível de interferência entre elas e cada pacote contém um indicador da rota para estabelecimento da mesma. Após o processamento, as rotas definidas tendem a reduzir ao máximo o nível de interferência entre elas.

Alguns trabalhos na literatura apresentam a integração de mais de duas camadas, como em [Wang & Wang, 2004] onde os autores apresentam o POWER (*Power On With Elected Rotation*). Esse é um protocolo que integra as camadas física, de enlace e de rede com o objetivo de determinar um número ótimo de agrupamentos em RSSFs, reduzindo o consumo de energia e melhorando a cobertura na rede. Um dos princípios desse protocolo é alternar o papel de nó lide com todos os nós do agrupamento. A priori,

o trabalho supõe que uma rede com poucos agrupamentos influenciará no consumo do nó líder, sobrecarregando o mesmo com um maior número de transmissões em virtude da grande área que ele deverá cobrir. Com isto, as três camadas trabalham em função de determinar a melhor quantidade de grupos que a rede terá.

Para isto, o algoritmo divide o funcionamento da rede em agrupamentos, com cada ciclo possuindo duas fases: a fase de construção da tabela de rodízio (CTP) e a fase onde o rodízio é efetuado e os dados sensorizados são transmitidos (RRP). A RRP é dividida em pequenos intervalos onde cada nó será eleito como o nó representativo de acordo com a tabela, conforme ilustra a figura 2.12. Assim, o protocolo MAC (POWER MAC) é adaptado para permanecer em modo escuta enquanto o nó em questão é o representativo do grupo, o que influi em maior custo de energia. Por isto o rodízio se torna uma abordagem que proporciona um consumo eficiente da energia. Resultados mostram que a quantidade de agrupamentos na rede favorece a recepção e a transmissão. Portanto, é importante escolher esse valor de acordo com aplicação para a qual a rede será projetada.

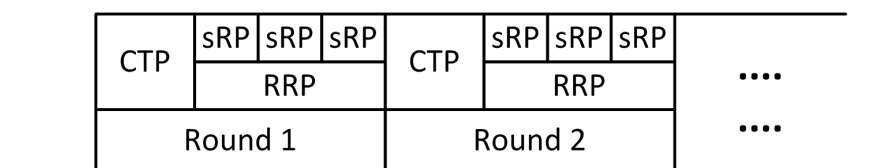


Figura 2.12. Procedimento de escalonamento do protocolo POWER [Wang & Wang, 2004].

Seguindo a abordagem do projeto integrado aplicado a mais de uma camada, o trabalho proposto em [Ding et al., 2003] apresenta um *framework* unificado que engloba a organização da rede, controle de acesso ao meio (MAC) e o protocolo de roteamento. O objetivo é utilizar de protocolos que são eficientes no consumo de energia, escaláveis e são protocolos simples. Este é um *framework* projetado para a arquitetura MINA (*Multi-hop Infrastructure Network Architecture*), que divide a rede em regiões de acordo com a quantidade de saltos que cada nó se encontra em relação ao nó *sink*. Com isto, os nós que possuem a mesma quantidade de saltos pertencerão à mesma região. Esta infra-estrutura possui centenas de nós sensores de baixa capacidade e alguns nós com capacidade maior que os nós móveis portáteis. No geral, o *framework* divide o funcionamento da rede em duas fases: auto organização da rede, na qual os nós descobrem seus vizinhos e trocam seus parâmetros (energia residual, número de saltos, tamanho de buffer disponível) e também reconhecem a topologia da rede local; e a fase de transmissão de dados na qual os nós transmitem e recebem dados para serem encaminhados para o nó *sink*. Essa fase envolve o protocolo MAC acessando o canal

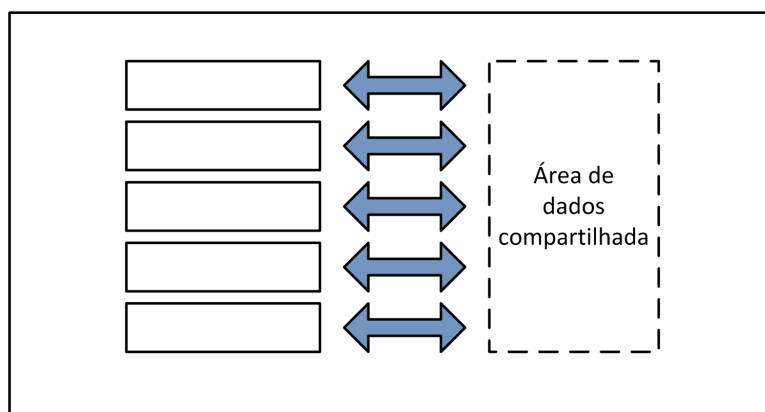


Figura 2.13. Modificação na arquitetura tradicional de camadas.

e o protocolo de roteamento para controlar o envio. Ambos trabalharão em função do contador de saltos de cada nó, estipulado pelo protocolo de organização da rede. O roteamento será efetuado com o encaminhamento dos pacotes para os nós de menor contador e o acesso ao meio será alocando e dividindo os canais para os nós de mesmo contador. Resultados mostraram que este tipo de organização funciona para poucos e muitos nós, sem apresentar problema algum.

2.6.3 Modificação na Arquitetura Tradicional

Outra maneira de se realizar um projeto integrado é propondo uma nova modificação na arquitetura de camadas atual. Esta modificação pode ser com a inclusão de mais algum módulo, *middleware* ou mesmo a adição de uma área de dados compartilhada com todas as outras. Os trabalhos descritos nesta seção apresentam uma análise da inclusão de uma área de memória compartilhada que irá intermediar a troca de informações entre as camadas. Esta área disponibiliza duas operações básicas: escrita e leitura de informação, e também mantém a organização tradicional de camadas, conforme é ilustrado na figura 2.13.

Em [Borgia et al., 2006], é proposto o MobileMAN (*Mobile Metropolitan Ad hoc Network*), um projeto financiado por um fundo de pesquisa da Comunidade Européia, com o objetivo de investigar as potencialidades das redes *ad hoc* em sua definição, projeto e avaliação de um estudo de caso usando as redes *ad hoc* em cidades. Neste projeto, é proposto uma arquitetura de camadas com um componente central, denominado *Network StatusNeSt*, atuando como um repositório das informações que serão compartilhadas pelos protocolos pertencentes a diferentes camadas de rede.

Nessa arquitetura, a interação entre as camadas é limitada aos parâmetros que são disponibilizados no componente central. A inovação da arquitetura é uma estrutura de

memória compartilhada (NeSt) que é um repositório das informações coletadas pelos protocolos de rede. Todos os protocolos projetados sobre a arquitetura podem acessar o componente central para escrever suas informações e para ler as informações produzidas e coletadas por outros protocolos. É importante destacar que essa arquitetura necessita da adaptação e o projeto de novos protocolos orientados a ela, a fim de usar todos os recursos disponíveis.

Em [Souto et al. \[2007\]](#), é proposto um trabalho que discute arquiteturas de projeto integrado existentes e propõe uma nova arquitetura de rede, denominada Arquitetura de Rede Orientada a Aspecto (*Aspect-Oriented Network Architecture*). Ao invés da comunicação entre camadas ser estritamente entre camadas adjacentes, essa arquitetura permite a comunicação direta entre quaisquer protocolos com a introdução de um novo mecanismo de abstração, o aspecto. Esse mecanismo permite à arquitetura de rede proposta o compartilhamento de informações e, ao mesmo tempo, garante a modularização de interesses que estão dispersos nas diferentes camadas como a energia, segurança, o gerenciamento de falhas e outras características relacionadas com parâmetros de QoS.

2.7 Conclusões

Um dos principais objetivos das RSSFs é transportar dados sensorizados de uma determinada região até o usuário final através de uma rota. Essa tarefa, também conhecida como comunicação de dados, engloba as fases de transmissão e recepção. As camadas envolvidas nesse processo são a camada de enlace e rede. A camada de enlace controla os modos de operação do rádio e qual nó acessa o meio para transmitir dados. A camada de rede estabelece a rota entre a origem e o destino dos dados. Assim, diversos protocolos que foram apresentados neste capítulo são propostos com o objetivo de promover uma comunicação eficiente para os nós nesse tipo de rede. Entretanto, esses protocolos não foram projetados considerando os requisitos das aplicações a serem sensoriadas, o que impede um melhor desempenho dos mesmos.

Neste trabalho propõe, portanto, um protocolo para a camada de enlace SMAC-DCC que faz uso das informações advindas da camada de rede para aperfeiçoar o seu funcionamento e melhorar a comunicação entre os nós sensores. O capítulo a seguir apresenta o funcionamento e as principais características do protocolo SMAC-DCC.

S-MAC COM CICLO DE TRABALHO DINÂMICO

Este capítulo apresenta o *Sensor-MAC with Dynamic Duty Cycle* (SMAC-DDC), um protocolo da camada MAC que adapta o *duty cycle* do nó para coleta de dados em RSSFs. Esse protocolo se baseia no funcionamento do tradicional protocolo encontrado na literatura *Sensor-MAC* [Ye et al., 2002]. A idéia principal é ajustar a duração do *duty cycle* dos nós sensores pertencentes à rota entre o nó *sink* e o nó fonte a partir de informações advindas da camada de rede. De uma forma geral o compartilhamento de informações entre camadas é restrito, surgindo então a necessidade de violar a arquitetura fixa de camadas criando um projeto integrado entre as camadas. Como visto na figura 3.1, os protocolos utilizados neste trabalho foram o *Directed Diffusion* na camada de rede e o S-MAC na camada de enlace e a troca de informações entre essas camadas ocorre de cima para baixo, com a camada de enlace recebendo informações da camada de rede na pilha de protocolos.

Nas seções seguintes serão apresentadas as principais características do protocolo SMAC-DDC. Na seção 3.1, é feita uma breve descrição dos protocolos utilizados e como ocorre a integração entre eles. A seção 3.2 explica como a adaptação do *duty cycle* de acordo com o fluxo de dados é feita. Em seguida, na seção 3.3 é explicado como o SMAC-DDC altera o ciclo de funcionamento do nó. Por fim, na seção 3.4 é apresentada uma breve síntese deste capítulo, apresentando as principais características desse novo protocolo (SMAC-DDC).

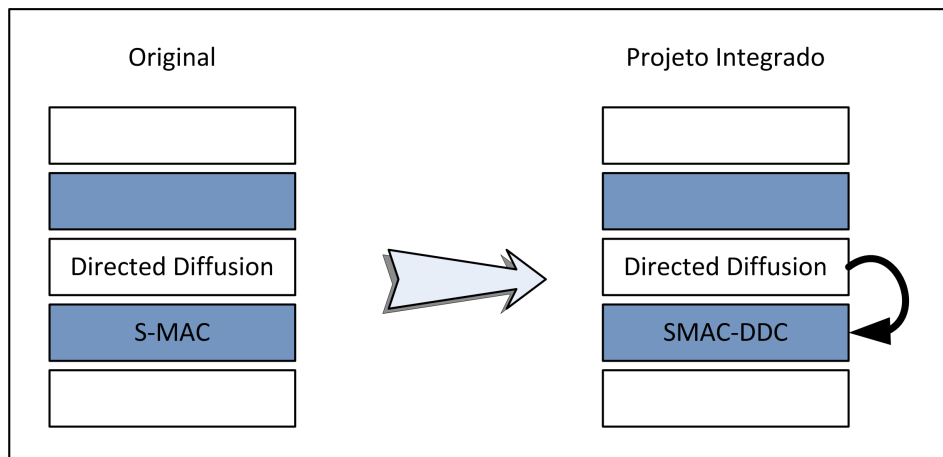


Figura 3.1. Compartilhamento de informação entre camadas.

3.1 Integração entre S-MAC e Directed Diffusion

Para avaliar o real benefício de um projeto integrado em RSSFs entre as camadas de enlace e de rede foram escolhidos dois protocolos específicos para esse tipo de rede. Na camada de rede, o protocolo escolhido foi o DD (*Directed Diffusion*) [Intanagonwiwat et al., 2000]. Esse é um protocolo de roteamento centrado em dados, no qual todos os nós estão cientes do tipo de dado trafegado. Além de criar uma rota entre o nó fonte e o nó *sink*, é permitida também a inserção de pontos de agregação de dados ao longo dessa rota. Neste trabalho utiliza-se a versão reativa desse protocolo conhecida por *PUSH* [Heidemann & Silva, 2003]. Essa versão elimina do DD a etapa de disseminação de interesses na rede (ver tabela 2.1). A criação da rota só é iniciada a partir do momento que o primeiro evento é sensoriado na rede. Esse é um protocolo de roteamento bastante conhecido na literatura, com o desenvolvimento voltado especificamente para atender as necessidades das RSSFs. Além disso, é um protocolo de roteamento desenvolvido para uma rede na qual o dado é entregue ao observador somente quando o evento é sensoriado, ou seja, um protocolo orientado a eventos.

Para a camada MAC optou-se por utilizar o protocolo S-MAC [Ye et al., 2002], um dos primeiros protocolos desenvolvidos para esse tipo de rede e muito referenciado na literatura. Além disso, esse protocolo apresenta algumas desvantagens, como exemplo, a definição dos períodos de dormência e trabalho de maneira fixa e invariável, impedindo com que o nó adapte o seu funcionamento à variabilidade do fluxo de dados na rede. Outra desvantagem apresentada pela inclusão dos constantes períodos de dormência para economia de energia a latência entre o nó origem e nó destino é aumentada, visto que no período de dormência o nó não poderá transmitir nenhum

quadro.

A latência está linearmente ligada a quantidade de saltos que o pacote faz na rede [Ye et al., 2004]. Todavia, com a utilização do S-MAC, os períodos de dormência contribuem para o aumento da latência em cada salto. Isso ocorre porque o nó sensor transmitirá os quadros somente no período em que o rádio estiver ligado. Logo, só é permitindo aos nós a realização de uma única transmissão por ciclo de funcionamento.

O aumento da latência gerado com o uso do protocolo S-MAC pode ser crucial em muitas aplicações devido à urgência na entrega das informações para o observador da rede. Nesse cenário, este trabalho propõe reduzir essa métrica na rede através da adaptação do *duty cycle* dos nós pertencentes à rota entre a fonte e o destino da informação. Essa adaptação é proporcionada pela troca de informações entre as duas camadas: enlace e rede. A seguir será descrito, passo a passo, como a adaptação do *duty cycle* consegue reduzir a latência ao mesmo tempo em que o consumo de energia é atenuado.

3.2 Adaptação do *Duty Cycle*

Nesta seção será apresentado a maneira como ocorre a adaptação do *duty cycle* propiciada pelo compartilhamento de informações entre as camadas de rede e de enlace. Inicialmente o *duty cycle* dos nós sensores é configurado através de um parâmetro α^1 . Conforme definido na tabela 3.1, no protocolo SMAC-DDC o valor do *duty cycle* dos nós no início de operação da rede é $\alpha/2$ pois, nesse momento os nós não pertencem a nenhuma rota. Configurado o *duty cycle* inicial, os nós iniciam o período de descoberta de vizinhos. Nesse momento eles trocam entre si pacotes de sincronização de suas agendas, por isso é necessário que todos os nós da rede permaneçam com seus rádios ligados para enviar e receber dos seus vizinhos quadros de sincronização. Dependendo da quantidade de vizinhos que cada nó possui essa fase pode ser bastante longa, não devendo ocorrer qualquer outra atividade na rede nesse momento.

Valor do <i>duty cycle</i>	S-MAC	SMAC-DDC
Nós não pertencentes à rota	α	$\alpha/2$
Nós pertencentes à rota	α	2α

Tabela 3.1. Valores de *duty cycle* definidos para os protocolos.

¹Parâmetro de entrada que define o valor do *duty cycle*, utilizado para viabilizar a comparação entre os protocolos S-MAC e SMAC-DDC.

Obedecendo à fase de descoberta de vizinhos do protocolo SMAC-DDC, o protocolo DD permanecerá ocioso sem a troca de qualquer informação até o término da mesma. A aplicação também fica impossibilitada de iniciar a coleta de dados de qualquer evento na rede. Após esse período inicial de sincronização e configuração, os nós sensores estão aptos para sensoriar e enviar os dados.

Terminado o período de descoberta de vizinhos do SMAC-DDC, a rede está pronta para iniciar a coleta de dados e encaminhá-los ao nó *sink*. Nesse momento, logo que um evento é detectado na rede, o protocolo DD inicia o processo de roteamento e estabelecimento das rotas conforme retratado na figura 3.2.

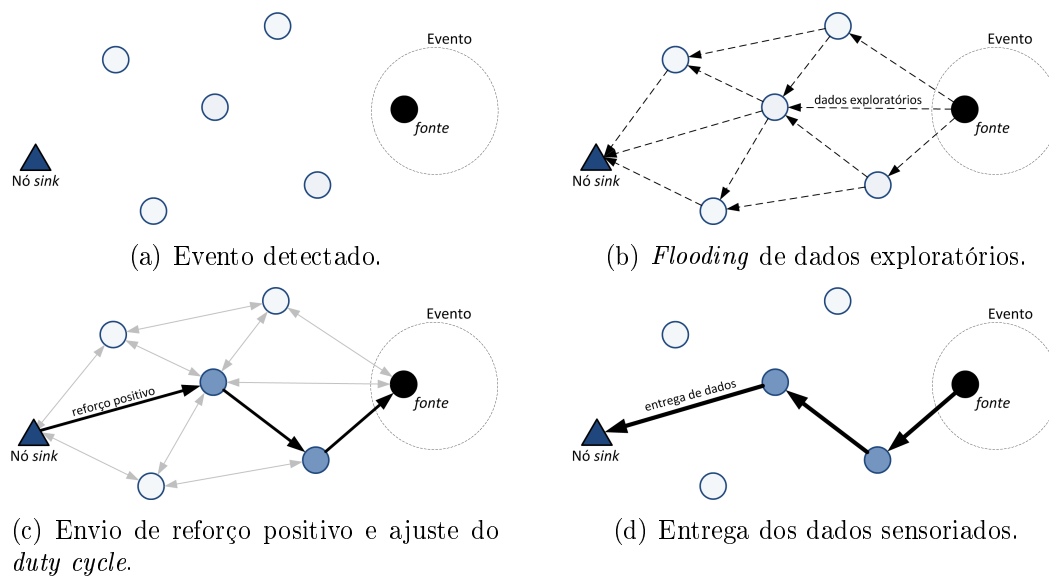


Figura 3.2. Etapas de funcionamento do Projeto Integrado.

Quando um evento é detectado na rede (figura 3.2-(a)), o nó que o detectou inicia uma inundação com pacotes do tipo dados exploratórios até alcançar o nó *sink* (figura 3.2-(b)). Ao receber o primeiro pacote com informações do evento, o nó *sink*, retorna um pacote do tipo reforço positivo pelo mesmo caminho que veio esse pacote. Assim, os nós que recebem um reforço positivo pertencem a rota entre a origem dos dados e o nó *sink* (figura 3.2-(c)).

O algoritmo 1 apresenta o trecho de código no qual o protocolo SMAC-DDC utiliza informações do protocolo DD. Conforme descrito, quando o nó recebe um pacote, o protocolo SMAC-DDC inspeciona o cabeçalho da camada de rede na camada de enlace para verificar de qual tipo é o pacote (linha 1). Sendo do tipo reforço positivo uma *flag* é habilitada indicando que aquele nó faz parte de uma rota (linha 2). Essa *flag* indica que, por pertencer à rota, o nó deve se comportar de maneira diferente dos demais, permanecendo com o seu rádio ligado por um tempo maior. Além disso, o nó deverá

informar aos seus vizinhos o seu novo agendamento, pois, com esse valor seus vizinhos saberão o momento certo para trocar os pacotes. Assim, toda vez que um nó muda o seu estado na rede será necessário o envio de pacotes para sincronização (linhas 3 e 6).

Após a configuração da rota, os dados são coletados e enviados para o nó *sink* em uma comunicação com múltiplos saltos, conforme ilustra a figura 3.2-(d). Nesse momento, todos os nós sensores que estiverem na rota entre a origem do evento e o nó *sink* estão com o seu *duty cycle* reconfigurado para 2α , conforme é mostrado na tabela 3.1, proporcionando uma entrega de dados mais rápida. Os demais nós da rede que não estiverem na rota permanecem com o valor do *duty cycle* inalterado.

Seguindo o funcionamento do protocolo DD, quando não houver mais atividade em alguma rota ou mesmo outro evento for detectado, surge a necessidade de atualizar algumas rotas. Nesse instante um pacote do tipo reforço negativo é enviado pelo nó *sink* para reconfiguração das rotas. Ao receber esse pacote, o protocolo SMAC-DDC desabilita a *flag* que indica se o nó está ou não na rota de acordo com o algoritmo 1 (linhas 4 e 5). A partir desse momento, o nó volta a seguir os valores iniciais de *duty cycle*, dormindo e acordando conforme os tempos pré-definidos.

Algoritmo 1: SMAC-DDC – Recebendo um pacote

```

input: Chega um pacote ao nó sensor

//Protocolo SMAC-DDC acessa no pacote as informações da camada
de rede
1 if tipoPacote = REFORÇO POSITIVO then
    | //Nó pertence à rota
    2 | pertencaRota = true;
    3 | sincronizarAgenda();
4 else if tipoPacote = REFORÇO NEGATIVO then
    | //O Nó deixa de pertencer à rota
    5 | pertencaRota = false;
    6 | sincronizarAgenda();
7 end

```

3.3 Pontos de Controle

Além do protocolo SMAC-DDC definir o funcionamento do nó em vários *duty cycle*, alguns pontos de controle são inseridos dentro de cada ciclo com o objetivo de habilitar a adaptação do *duty cycle* dos nós. Nesses pontos de controle será definido o funcionamento de cada nó de acordo com o *status* de pertencer ou não à rota de entrega de dados do nó fonte para o nó *sink*.

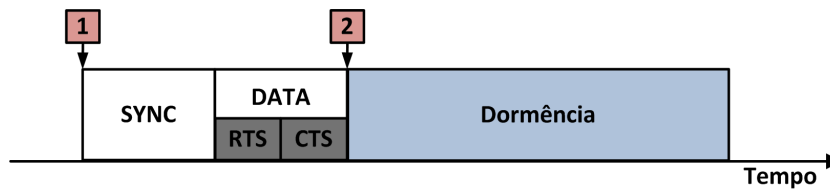


Figura 3.3. Pontos de Controle.

A figura 3.3 indica o momento em que esses pontos de controle são executados. No período de SYNC, o nó deve enviar pacotes de sincronização de agenda para os seus vizinhos. Esses pacotes indicarão o próximo instante que o nó estará acordado. Normalmente, a periodicidade desse envio é calculada de acordo com a quantidade de vizinhos que o nó possui. Sendo uma rede muito densa, o intervalo entre o envio de pacotes SYNC deverá ser menor. Entretanto, quando o nó faz parte de alguma rota ele deve enviar mais pacotes de sincronização com a finalidade de avisar aos seus vizinhos sua nova agenda. Assim, ao chegar ao ponto de controle 1 o nó verifica se ele faz ou não parte da rota. Caso o nó pertença à rota ele assume a nova agenda e envia um pacote de sincronização para os seus vizinhos.

Terminado o *duty cycle* e não havendo nenhuma comunicação *unicast* iniciada, o nó sensor desliga o seu rádio para economizar energia. No entanto, o tempo que ele permanece com o seu rádio desligado será diferenciado de acordo com o estado que o nó se encontra. Por isso, no ponto de controle 2 o nó verifica qual é o seu estado. Se o seu estado for pertencente à rota, ele deve dormir um tempo menor. Caso negativo o tempo de dormência será maior. Esses tempos são definidos em função do *duty cycle* do nó, ou seja, quanto maior for o *duty cycle* do nó, menor será o tempo que o mesmo permanecerá no estado de dormência. Assim, o nó define o seu tempo de dormência de acordo com o seu estado e desliga o rádio durante esse tempo. Após o período de dormência, o nó acorda e um novo ciclo de funcionamento é iniciado.

3.4 Conclusão

O presente capítulo apresentou o protocolo SMAC-DDC (*Sensor-MAC Dynamic Duty Cycle*), um protocolo para coleta de dados em RSSFs. Esse protocolo utiliza a técnica de projeto integrado entre as camadas de enlace e rede para ajustar o *duty cycle* dos nós sensores em função do fluxo de dados na rede. A idéia principal do SMAC-DDC é a camada MAC se beneficiar das informações advindas da camada de rede para ajustar o *duty cycle* de forma a minimizar a latência e reduzir o consumo de energia da rede, sem prejudicar a quantidade de dados entregues ao nó *sink*. No próximo capítulo serão

apresentados os resultados de simulação para o algoritmo proposto neste capítulo.

RESULTADOS DE SIMULAÇÃO

O presente capítulo apresenta os resultados obtidos com a simulação do protocolo SMAC-DDC em comparação com o protocolo S-MAC em cenários para coleta de dados em RSSFs. Diversos eventos foram iniciados de diferentes maneiras na rede e a coleta iniciada. Para viabilizar essa coleta, utilizou-se o protocolo *Directed Diffusion* na camada de rede que cria caminhos para roteamento dos dados entre o nó fonte e o nó *sink*. Assim, o objetivo é avaliar o comportamento dos protocolos SMAC-DDC e S-MAC com o protocolo *Directed Diffusion*, conforme ilustra a figura 4.1, e quantificar o ganho alcançado na coleta de dados com a adaptação do funcionamento do nó sensor de acordo com o fluxo de dados na rede. Essa adaptação é realizada através do compartilhamento de informações entre a camada de rede e a camada de enlace.

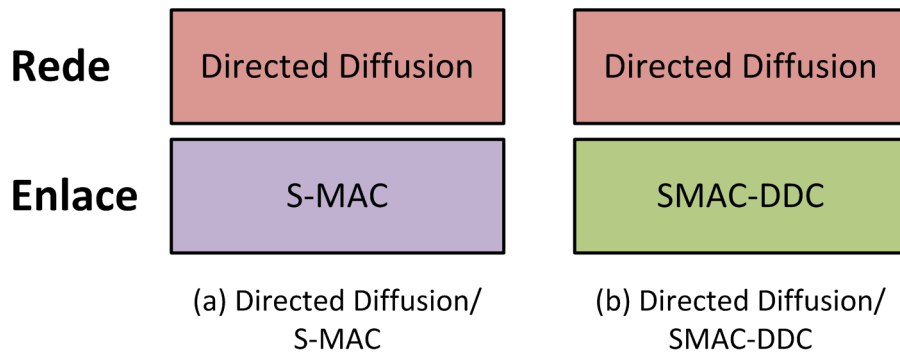


Figura 4.1. Protocolos dos cenários avaliados.

A organização deste capítulo é disposta da seguinte maneira: a seção 4.1 apresenta os cenários de simulação utilizados na execução dos algoritmos. Além disso, define os valores padrão para os parâmetros de simulação e também a forma como os eventos ocorrem na rede através de distribuições estatísticas. Em seguida, a seção 4.2

apresenta um projeto experimental $2^k r$ que mensura o impacto das mudanças de alguns parâmetros de rede no valor final da latência, quando se utiliza o protocolo S-MAC que alterna os modos de operação do rádio. Em seguida, na seção 4.3, é feita uma avaliação do funcionamento correto do protocolo SMAC-DDC através da captura de mapas de *duty cycle* em diferentes instantes de simulação. Nas seções seguintes, o desempenho do protocolo SMAC-DDC com o protocolo S-MAC é comparado utilizando algumas métricas. Na seção 4.4 são apresentados os resultados da latência média acumulada. Essa métrica determina o tempo médio entre o envio de um pacote pelo nó fonte até o nó *sink*. Em seguida, na seção 4.5, o consumo de energia dos dois protocolos é avaliado e comparado, e, na seção 4.6, são apresentados os resultados relativos a taxa de entrega de mensagens ao nó *sink* pelos protocolos durante toda a simulação. Por fim, a seção 4.7 apresenta uma breve conclusão dos resultados.

4.1 Cenário de Simulação

O cenário para as simulações considera que os sensores são depositados em uma área formando uma topologia plana de $100 \times 100 m^2$, com 200 nós posicionados aleatoriamente na rede, exceto o nó *sink* que está no canto inferior esquerdo e possui posição fixa $(0, 0)$. Não foi considerada nenhuma mobilidade para os nós e todos eles possuem o mesmo *hardware* caracterizando a rede como homogênea. Cada nó sensor possui uma energia inicial de 25 J e o modelo do consumo de energia utilizado é definido pelo simulador que decrementa o gasto de energia de acordo com o modo de operação que o rádio do nó sensor se encontra. Cada modo de operação tem o seu consumo definido e, neste trabalho, o consumo foi configurado de acordo com o nó sensor modelo Mica2 [Crossbow, 2008], conforme definido na tabela 4.1. Todas as simulações foram feitas no simulador *Network Simulator 2.32* [ns2, 2002] e o tempo de simulação é 1200 s. Os resultados obtidos correspondem à média aritmética de r simulações, em que escolheu-se um r de tal forma que o intervalo de confiança seja de 95% dos valores [Jain, 1991].

Durante a simulação, definiu-se um intervalo para que ocorram eventos. Os eventos iniciam na rede após o período de descoberta de vizinhos estabelecido pelos protocolos S-MAC e SMAC-DDC, no instante 70 s. A geração de eventos será interrompida no instante 800 s. Assim, a partir desse momento os nós cooperarão para entrega dos últimos pacotes ao nó *sink* até o final da simulação (instante 1200 s). A priori, a geração dos eventos foi definida de duas maneiras: eventos fixos e eventos aleatórios. A tabela 4.1 apresenta os parâmetros de cada evento. Para os eventos fixos é definida

Parâmetros	
Simulador	<i>Network Simulator 2.32</i>
Topologia da rede	<i>plana</i>
Nós (<i>hardware</i>)	Homogêneos
Dimensão da rede	100 × 100 m ²
Quantidade de nós	200
Raio de comunicação	15 m
Energia inicial	25 J
Instante inicial dos eventos	70 s
Instante final dos eventos	800 s
Eventos Fixos	
Distância mínima ao nó <i>sink</i>	80 m
Duração	70 s
Raio do evento	10 m
Eventos Aleatórios	
Raio do evento mínimo	2 m
Raio do evento máximo	10 m
Duração do evento mínimo	5 s
Duração do evento máximo	50 s
Valor do parâmetro λ	0.03
Consumo de Energia do Rádio	
Transmissão de dados	27 mA
Recepção	10 mA
Estado de dormência	1 μ A
Potência	+5 dBm

Tabela 4.1. Parâmetros de simulação utilizados.

uma duração fixa e o raio de influência do evento é de 10 m, garantindo que haverá pelo menos um nó no raio desse evento. Esses eventos chegam à rede de forma fixa, um após o outro, após o instante 70 s. A quantidade desses eventos é determinada pela variável σ . Essa variável também determina o intervalo entre os eventos fixo, quanto maior for o σ , menor será o intervalo de tempo entre os eventos. Definiu-se também para esses eventos uma distância mínima em relação ao nó *sink*. Essa distância tem como objetivo estimular na rede eventos em regiões com altos valores de latência, retirando a interferência dos eventos que ocorrem muito próximo ao nó *sink*. Conforme ilustrado na figura 4.2, os eventos fixos ocorrerão apenas na região mais escura.

Os eventos aleatórios podem acontecer em qualquer lugar dentro das dimensões da rede, não obedecendo a nenhuma restrição. O raio de influência de cada evento é um valor dentro do intervalo definido entre *raio do evento mínimo* e *raio do evento*

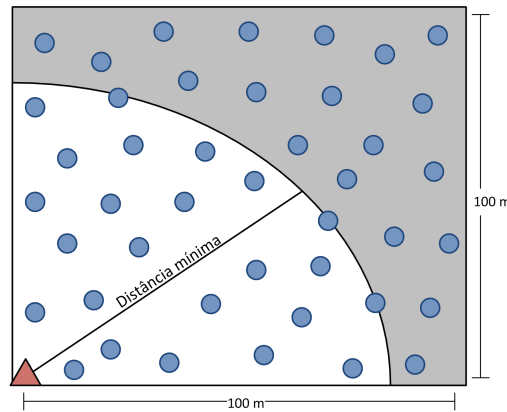


Figura 4.2. Distância mínima para ocorrência dos eventos fixos.

máximo e a duração máxima de cada evento é 50 s. Eles chegam à rede por uma distribuição de Poisson com parâmetro λ [Ross, 1996]. Nesse modelo os eventos acontecem aleatoriamente e são independentes um do outro. O número de eventos em cada segundo de simulação é descrito por uma variável aleatória:

$$P(X = x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

4.2 Projeto Experimental

Nesta seção é descrito o modelo do projeto experimental utilizado para avaliar o impacto que alguns parâmetros de rede exercem sobre o valor final da latência. Dependendo do valor de cada parâmetro, a latência pode ser alta e inviabilizar a coleta de dados na rede. Além disso, conhecendo o impacto dos parâmetros no valor da latência, novos protocolos podem ser desenvolvidos criando técnicas que alteram esses parâmetros e, conseqüentemente, reduzem a latência na entrega de pacotes ao nó *sink*. Em [Ye et al., 2004], os autores apresentam uma análise da latência em protocolos MAC para RSSFs, na qual foram identificadas algumas fontes que geram atrasos na transmissão dos pacotes e influenciam o valor final da latência. Essas fontes são:

- *Detecção de portadora*: essa tarefa introduz um atraso na transmissão de pacotes, pois, antes de realizar qualquer transmissão o nó escuta o meio para verificar se há alguma atividade. O tempo que o nó realiza essa tarefa é determinado pelo tamanho da janela de contenção.

- *Atraso gerado pelo backoff*: quando a detecção de portadora falhar, quer seja pelo nó detectando outra transmissão, quer pela ocorrência de uma colisão, o nó aguardará um período de tempo aleatório para iniciar uma nova tentativa de transmissão.
- *Atraso na transmissão*: é determinado pelo tempo entre a requisição de um dado e a resposta positiva de seu recebimento. Esse atraso é determinado pela largura de banda do canal, tamanho do pacote e pelo esquema de codificação adotado.
- *Atraso na propagação*: a distância entre o nó emissor e o nó receptor determina o tempo de propagação entre esses nós. Portanto quanto maior for essa distância maior será o tempo para realizar a troca de dados. Em RSSF, a distância entre os nós é pequena e esse atraso pode ser desconsiderado.
- *Atraso no processamento*: ao receber um pacote, o nó precisa processá-lo antes de encaminhá-lo ao próximo nó. Entretanto, dependendo do poder de processamento do nó sensor e dos algoritmos para processamento de dados escolhidos, o tempo para processar o pacote pode ser grande e atrasar a entrega de dados.
- *Atraso pela fila*: é determinado pela quantidade de pacotes que estiverem na fila para o receptor processar o recebimento e depende da quantidade de tráfego na rede. Pode ser um fator dominante e impactar a latência.

A inclusão de ciclos periódicos de dormência pelo protocolo S-MAC incrementa o valor da latência. Para capturar o impacto real dos parâmetros no valor final da latência, utiliza-se um Projeto Experimental Fatorial do tipo $2^k r$, descrito em [Jain, 1991], onde k corresponde ao número de parâmetros escolhidos para avaliação, r representa a quantidade de replicações no experimento e 2 à quantidade de níveis que cada parâmetro pode assumir. Maiores detalhes sobre o cálculo desse projeto serão apresentados no Apêndice B.

Nesse projeto foram escolhidos três parâmetros considerados importantes e, que impactam a latência dos pacotes que chegam ao nó *sink*: raio de comunicação(R), *duty cycle*(D) e número de eventos(σ). O primeiro deles, o raio de comunicação, determina diretamente a quantidade de saltos que um pacote faz na rede. Esse valor influencia o valor final da latência porque quanto menor a quantidade de saltos menor será o valor da latência. Em cada salto um atraso é acrescido ao valor da latência e, para diminuir a quantidade de saltos, basta aumentar o raio de comunicação do nó. No entanto, a variação desse parâmetro pode aumentar a quantidade de colisões, o que aumentará a latência na transmissão dos pacotes. O segundo parâmetro avaliado foi o *duty cycle*

dos nós. Esse parâmetro influencia o valor da latência uma vez que determina o tempo em que o nó permanece com o seu rádio ligado. Assim, quanto menor for o *duty cycle*, menor será o tempo que o nó terá para receber e enviar dados na rede. Finalmente, o terceiro parâmetro foi o número de eventos na rede. Quanto maior for esse valor, maior será a quantidade de dados trafegados na rede. Conseqüentemente têm-se colisões e retransmissões de pacotes implicando em um aumento no valor da latência.

O protocolo MAC escolhido para avaliação das alterações da latência em função dos parâmetros selecionados foi o protocolo S-MAC [Ye et al., 2002] (veja seção 2.3). Na camada de rede utilizou-se o protocolo *Directed Diffusion* (veja seção 2.5). Para geração dos eventos escolheu-se o modelo de geração de eventos fixos. Os demais parâmetros são ajustados de acordo com os valores definidos na tabela 4.1.

O modelo para o cálculo da latência pode ser definido da seguinte forma:

$$\begin{aligned} \widehat{Latência} = & q_0 + q_\sigma x_\sigma + q_R x_R + q_D x_D + q_{\sigma R} x_\sigma x_R + \\ & q_{\sigma D} x_\sigma x_D + q_{RD} x_R x_D + q_{\sigma RD} x_\sigma x_R x_D + e \end{aligned}$$

onde q_R representa o efeito do raio de comunicação R exerce sobre o valor da latência, q_D representa o efeito que a variação do *duty cycle* exerce na latência e q_σ corresponde ao efeito da variação da quantidade de eventos no valor da latência. A interação entre os parâmetros tomados dois a dois são retratados pelos coeficientes $q_{\sigma R}$, $q_{\sigma D}$ e q_{RD} . O coeficiente $q_{\sigma RD}$ exprime a interação entre os três parâmetros e a constante e captura os erros no modelo. As variáveis x_σ , x_R e x_D assumem valores indicados pelos níveis 1 e -1 , conforme descrição a seguir:

$$x_\sigma = \begin{cases} -1 & \text{se a quantidade de eventos for 20} \\ 1 & \text{se a quantidade de eventos for 10} \end{cases}$$

$$x_R = \begin{cases} -1 & \text{se o raio de comunicação for 15} \\ 1 & \text{se o raio de comunicação for 25} \end{cases}$$

$$x_D = \begin{cases} -1 & \text{se o } \textit{duty cycle} \text{ for 10\%} \\ 1 & \text{se o } \textit{duty cycle} \text{ for 40\%} \end{cases}$$

A tabela 4.2 apresenta os resultados das simulações para cada configuração de parâmetros escolhida. Cada linha representa a latência média de acordo com as replicações para cada configuração definida pelos níveis de *fontes*, *raio* e *duty cycle*. A linha *%latência* mostra a porcentagem que a variação de cada parâmetro impacta no valor da latência.

<i>I(média)</i>	x_σ	x_R	x_D	$x_\sigma x_R$	$x_\sigma x_D$	$x_R x_D$	$x_\sigma x_R x_D$	latência
1	-1	-1	-1	1	1	1	-1	154,82
1	-1	-1	1	1	-1	-1	1	94,72
1	-1	1	-1	-1	1	-1	1	202,02
1	-1	1	1	-1	-1	1	-1	102,83
1	1	-1	-1	-1	-1	1	1	116,63
1	1	-1	1	-1	1	-1	-1	45,82
1	1	1	-1	1	-1	-1	-1	130,33
1	1	1	1	1	1	1	1	54,39
112,69	-25,90	9,70	-38,26	-4,13	1,57	-5,53	4,25	<i>latência</i>
-	25%	3%	55%	0%	0.5%	1%	0%	<i>%latência</i>

Tabela 4.2. Resultados do Projeto Experimental 2^3r .

Assim, após o cálculo dos coeficientes o modelo para calcular a latência fica definido da seguinte forma:

$$\begin{aligned}
 \text{Latência} = & 112,69 - 25,90x_\sigma + 9,70x_R - 38,36x_D - 4,13x_\sigma x_R + \\
 & 1,57x_\sigma x_D - 5,53x_R x_D + 4,25x_\sigma x_R x_D + e
 \end{aligned}$$

Com esse projeto foi possível avaliar que a variação do *duty cycle* impacta em 55% no valor da latência. Esse impacto pode ser observado na tabela, pois, as linhas de cor cinza apresentam o resultado das simulações nas quais o valor para o *duty cycle* era 10% e, logo abaixo de cada linha de cor cinza, é apresentado o resultado da mesma configuração com mudança apenas no valor do *duty cycle* para 40%. Pode-se notar que o valor da latência é reduzido toda vez que o *duty cycle* dos nós é aumentado. É visto também que o valor do coeficiente x_D na tabela (-38,26) é negativo. Isso significa que a variação do *duty cycle* de 10% para 40% contribui para uma expressiva diminuição da latência. Esse comportamento é apresentado na latência, pois quanto maior for o *duty cycle* dos nós, maior também será o tempo que cada nó escutará a rede, aumentando, portanto, a quantidade de pacotes recebidos e enviados em um mesmo intervalo de tempo.

Outro parâmetro que exerce influência no valor da latência é o σ que define a quantidade de eventos. A variação deste parâmetro de 20 para 10 impacta em 25% no valor da latência. Isso ocorre devido ao fato de que o σ determina a quantidade de eventos e a redução desse valor implica em uma taxa menor de dados trafegando na rede, diminuindo a latência na transmissão dos dados. Por último verifica-se que a variação do raio de comunicação impacta em 3% no valor da latência. Apesar de ser uma pequena influência, explica o fato de que o nó com raio maior transmitirá o pacote com uma quantidade menor de saltos. A interação entre os parâmetros também se demonstra desprezível, pois não exerce influência na latência. A porcentagem restante $100\% - 85\% = 15\%$ é atribuída aos erros experimentais e que não foram capturados pelas simulações e nem mesmo pelos valores escolhidos na variação dos parâmetros.

A partir desse projeto foi possível concluir que dentre os parâmetros escolhidos para variação, o que apresentou significativa influência no valor da latência é o *duty cycle*. Quanto maior o período de *duty cycle* do nó sensor, maior também será o tempo que o nó permanece com o seu rádio ligado para transmitir e enviar informações. A figura 4.3 apresenta um gráfico que mostra o impacto que a variação do *duty cycle* exerce sobre o valor da latência no protocolo S-MAC. Todos os parâmetros de simulação foram fixados com a variação apenas do *duty cycle*. Pode-se observar que à medida que o *duty cycle* aumenta, o valor médio da latência por pacote diminui, o que confirma os resultados obtidos pelo projeto experimental. Entretanto, é importante ressaltar a necessidade do estudo e projetos de protocolos MAC que reduzem a influência desse parâmetro no valor final latência, sem alteração do consumo de energia da rede.

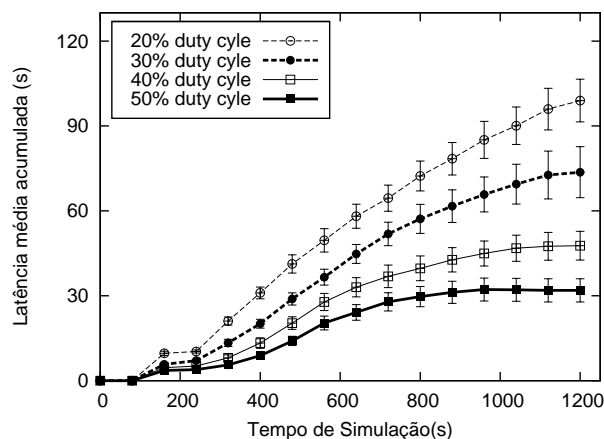


Figura 4.3. Latência média variando o *duty cycle* no protocolo S-MAC.

4.3 Funcionamento Básico

Nesta seção será feita uma avaliação do funcionamento do protocolo SMAC-DDC. Esse protocolo trabalha em cooperação com o protocolo de roteamento de dados *Directed Diffusion*, que estabelece uma rota entre o nó fonte e o nó *sink*. Para isso, conforme definido na seção 3.2, algumas etapas são seguidas até a concretização do caminho entre o emissor e o destino.

Para avaliar o funcionamento do protocolo SMAC-DDC proposto foi simulado a coleta de dados apenas com a geração de um evento na rede. O evento acontece respeitando a mínima distância ao nó *sink* de 80 m. Assim, a partir do momento que o evento é detectado na rede, os mapas dos *duty cycle* dos nós sensores são construídos, conforme ilustrado na figura 4.4, na qual o nó *sink* está na posição (0,0). Os demais nós com marca (□) possuem o *duty cycle* de 15% e os nós com marca (●) possuem o *duty cycle* de 60%, conforme as proporções definidas no funcionamento do protocolo SMAC-DDC.

Verifica-se que no instante em que o evento é detectado (figura 4.4-(a)) todos os nós possuem o mesmo *duty cycle*. Nesse momento apenas pacotes de dados exploratórios estão sendo enviados pelo *Directed Diffusion* com a finalidade de estabelecer a rota. No instante em que o primeiro pacote exploratório chega ao nó *sink*, um pacote de reforço positivo é enviado em direção ao evento na mesma direção que veio esse pacote exploratório. Assim, os nós que recebem o pacote de reforço positivo aumentam o valor de seu *duty cycle* e a rota é construída. Pode-se perceber esse comportamento nas figuras 4.4-(b), 4.4-(c) e 4.4-(d). Observa-se também o correto funcionamento do algoritmo, pois a criação da rota para envio dos dados continua progressivamente até chegar ao nó fonte, conforme ilustra a figura 4.4-(e).

4.4 Latência

Nesta seção será avaliada a métrica latência para os pacotes que chegam ao nó *sink*. Essa métrica é definida como o tempo entre a transmissão de um pacote pelo nó fonte e a recepção do pacote pelo nó destino. Conforme discutido na seção 4.2, quando se usa um protocolo que acrescenta ciclos periódicos de dormência esse valor tende a aumentar. O protocolo proposto neste trabalho tem como objetivo reduzir esse valor através da adaptação do *duty cycle* dos nós em função do fluxo de dados na rede. Nesta seção, os resultados obtidos com as simulações serão apresentados e discutidos.

A tabela 4.4 exibe os valores da latência do primeiro pacote a chegar ao nó *sink*

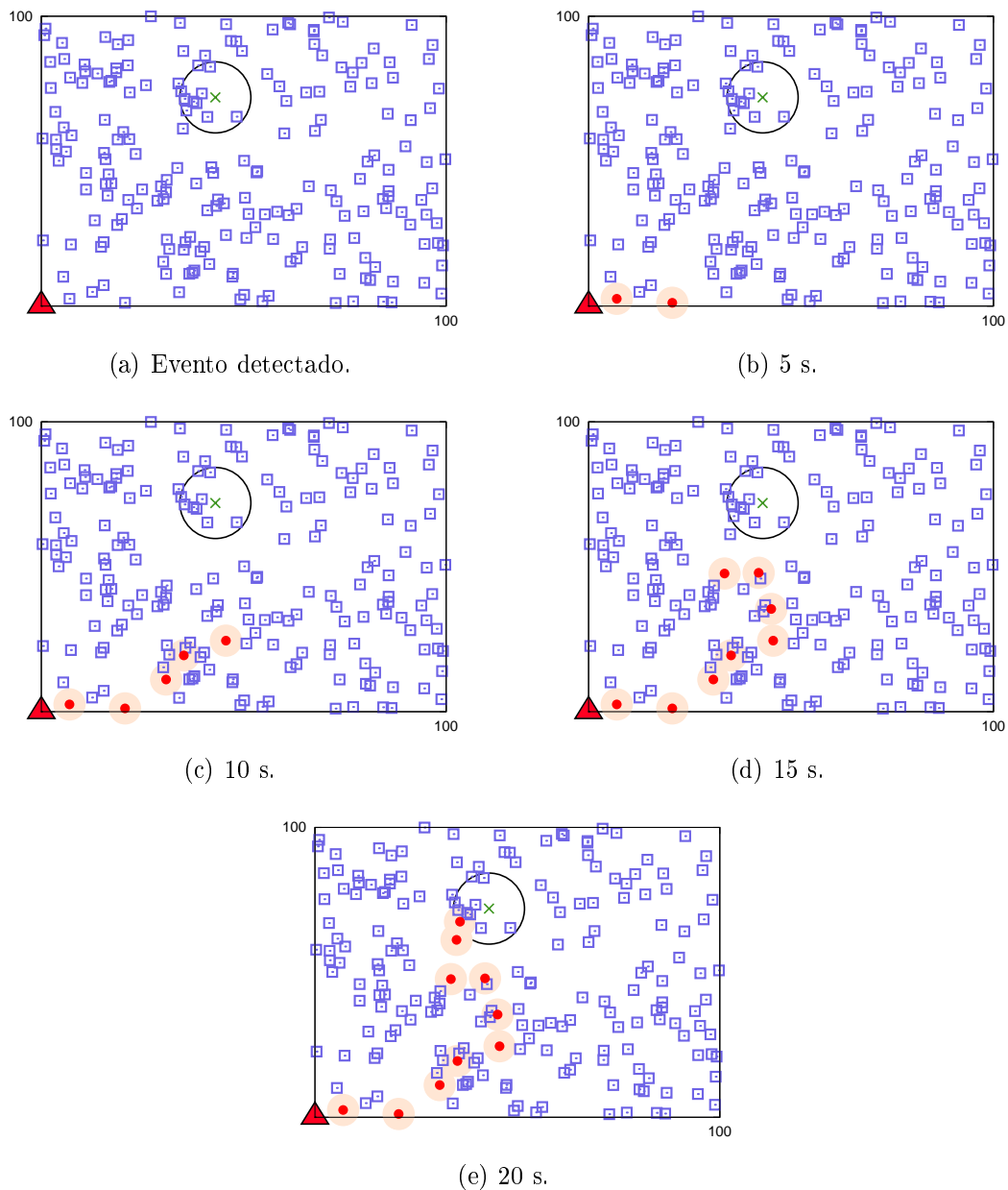


Figura 4.4. Mapa do *duty cycle* dos nós na rede em diferentes momentos.

para diferentes valores de α^1 . O funcionamento do protocolo SMAC-DDC define o *duty cycle* inicial dos nós em $\alpha/2$, assim pode-se observar que esse protocolo apresenta uma grande latência para o primeiro pacote chegar ao nó *sink*. Mesmo com o aumento do valor de α , o protocolo SMAC-DDC apresenta valores maiores do que o protocolo S-MAC. Essa diferença ocorre, pois, nos primeiros momentos de coleta de dados de um evento os nós pertencentes à rota estão sincronizando suas agendas. Após esse

¹Variável utilizada pelo protocolo SMAC-DDC para definir um limiar de configuração do *duty cycle* dos nós.

momento de sincronização, todos os nós na rota estarão com a agenda de seus vizinhos atualizada e o valor da latência reduzirá nas próximas transmissões realizadas. Esse comportamento não ocorre no protocolo S-MAC, pois, por terem o mesmo valor de α , os nós não precisam dessa nova sincronização de agendas quando um evento é detectado.

Protocolo	Média	Intervalo de confiança
SMAC $\alpha = 30\%$	6,07	(5,56 : 6,59)
SMAC-DDC $\alpha = 30\%$	12,95	(11,97 : 13,93)
SMAC $\alpha = 40\%$	4,84	(4,54 : 5,14)
SMAC-DDC $\alpha = 40\%$	9,67	(8,87 : 10,47)

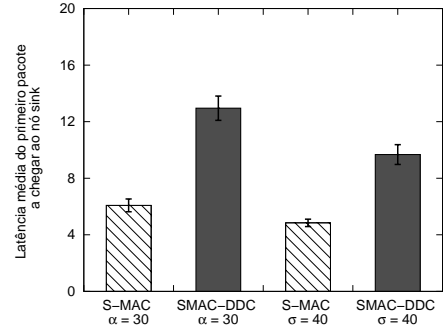
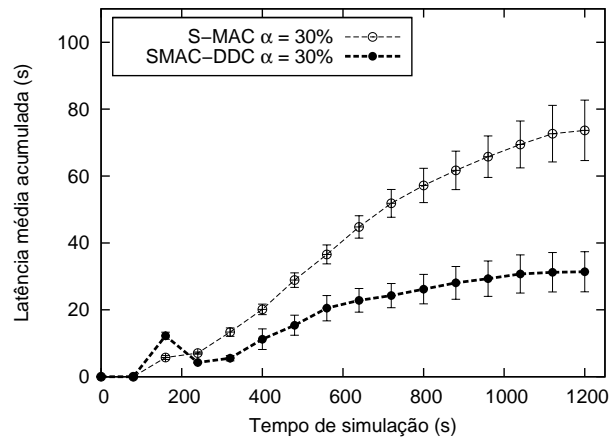
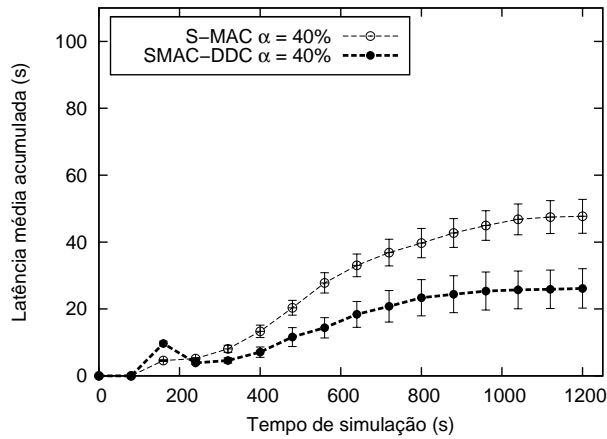


Tabela 4.3. Latência do primeiro pacote para diferentes valores de α .

A figura 4.5 apresenta os resultados da latência média acumulada durante toda a simulação com a variação de α em um cenário com eventos fixos ($\sigma = 10$). Pode-se observar que a solução proposta reduz em 57% (de 73,67 s para 31,4 s) o valor da latência quando o α é de 30% (figura 4.5-(a)) e em 45% (de 47,7 s para 26,14 s) quando o α é 40% (figura 4.5-(b)). Essa redução é alcançada, pois, o protocolo SMAC-DDC possibilita um número maior de transmissões em um mesmo intervalo de tempo. Assim pode-se constatar que com o aumento do *duty cycle* dos nós que farão parte do roteamento para 2α é possível reduzir consideravelmente o valor da latência. Nota-se também que a variação do α de 30% para 40% minimiza a diferença entre as duas abordagens, pois as curvas do protocolo SMAC-DDC sofrem uma pequena diminuição e as curvas do protocolo SMAC reduzem cerca de 35%.

Outro ponto importante a salientar na figura 4.5 são os picos apresentados na latência inicial do protocolo SMAC-DDC. Esses picos são ocasionados devido aos altos valores apresentados pelos primeiros pacotes que chegam ao nó *sink*, conforme foi mostrado na tabela 4.4. Nesse instante, o *duty cycle* dos nós estão sendo ajustados para 2α e suas agendas são atualizadas formando uma rota para a coleta de dados. Logo, após esse momento o valor da latência média acumulada reduz em função dos novos pacotes que chegam ao *sink* com baixo valor de latência.

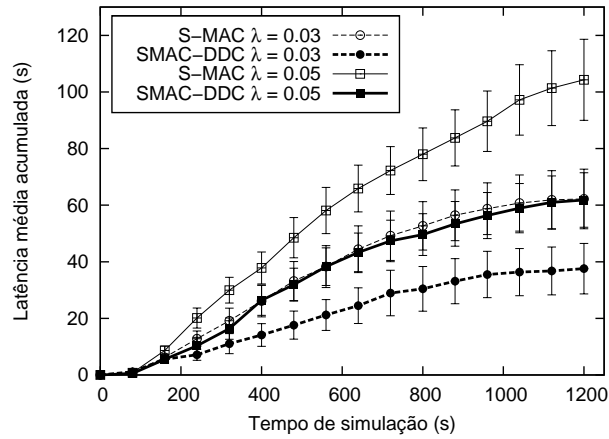
Dando seqüência a avaliação do protocolo SMAC-DDC, escolheu-se analisar o impacto da geração de eventos no comportamento dos protocolos escolhidos. Deste modo elegeram-se dois cenários para essa análise: um cenário de eventos aleatórios com o valor de λ de 0.03 e 0.05 e um cenário de eventos fixos variando a quantidade de eventos na rede σ entre 10 e 15. Para os dois cenários o valor do α foi fixado em

(a) $\alpha = 30\%$.(b) $\alpha = 40\%$.**Figura 4.5.** Comparação da latência para diferentes valores de α .

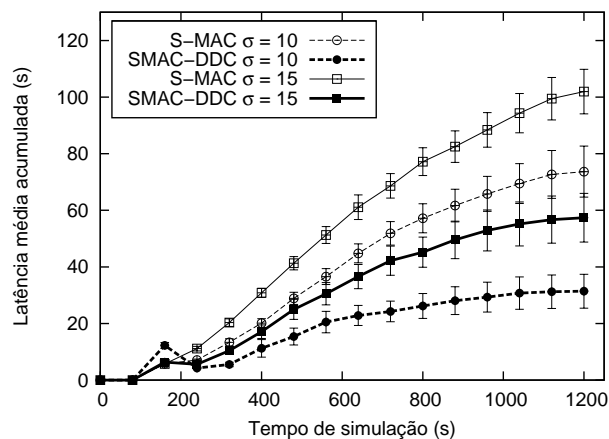
30%. A figura 4.6 apresenta os resultados comparativos. Pode-se observar no gráfico da figura 4.6-(a) que o protocolo SMAC-DDC reduz em média o valor da latência em 40% para ambos valores de λ . Além disso, nota-se uma tendência constante nas curvas do protocolo SMAC-DDC a partir do instante de 800 s no qual o último evento é detectado. Isso indica que, após esse momento, poucos pacotes chegam ao nó *sink* ou então, a latência dos pacotes que chegam é a mesma do valor médio. Por isso, o valor da latência permanece invariável. Esse comportamento não é verificado nas curvas do protocolo S-MAC, pois se percebe uma tendência crescente nessas curvas indicando que mesmo tendo terminado o último evento alguns pacotes retardatários chegam ao nó *sink*. Destaca-se também no gráfico que as curvas do SMAC-DDC com $\lambda = 0.05$ e do S-MAC com $\lambda = 0.03$ coincidem, indicando que para uma carga maior de dados o SMAC-DDC apresenta um valor de latência semelhante ao S-MAC quando sujeito a um tráfego menor. Isso reforça a conjectura de que o protocolo SMAC-DDC apresenta

um melhor desempenho no quesito latência quando comparado ao protocolo S-MAC.

A figura 4.6-(b) exibe os resultados para um cenário de eventos fixos. Observa-se que o protocolo SMAC-DDC apresenta um pico na latência no início da coleta de dados. Isso ocorre, pois, enquanto as agendas dos nós sensores que pertencem à rota não são sincronizadas, os dados iniciais dos eventos chegam ao nó *sink* com um alto valor de latência. Entretanto esse valor é maior, sendo mais perceptível a diferença nesse cenário devido à distância mínima que os eventos obedecem. Essa restrição incrementa o valor da latência gerado pela propagação do pacote de dados. Observa-se ainda que a redução na latência proporcionada pelo protocolo SMAC-DDC é, em média, de 50% quando comparado ao protocolo S-MAC, independente da quantidade de eventos. Com isso, verifica-se que o protocolo SMAC-DDC provê uma redução significativa na latência dos pacotes que chegam ao nó *sink* em um cenário de eventos fixos.



(a) Diferentes valores de λ para eventos aleatórios.



(b) Diferentes valores de σ para eventos fixos.

Figura 4.6. Comparação da latência para diferentes formas de evento.

Em todos os gráficos que apresentam resultados da latência é observado que, para

os protocolos escolhidos o valor da latência aumenta durante o decorrer da simulação. Isso ocorre, pois, novos eventos chegam à rede e os primeiros pacotes de cada evento são entregues ao nó *sink* com valor de latência maior do que os demais devido ao estabelecimento das rotas por parte do protocolo DD.

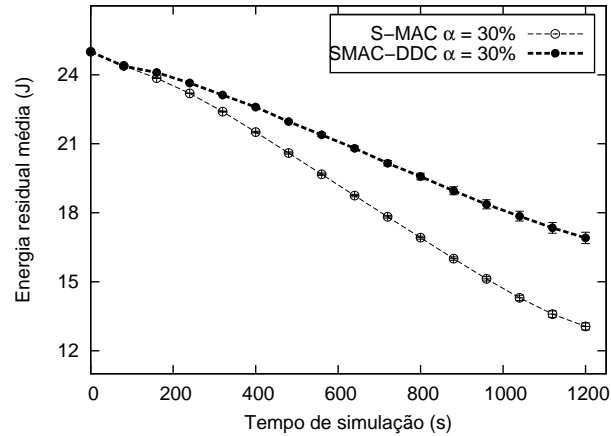
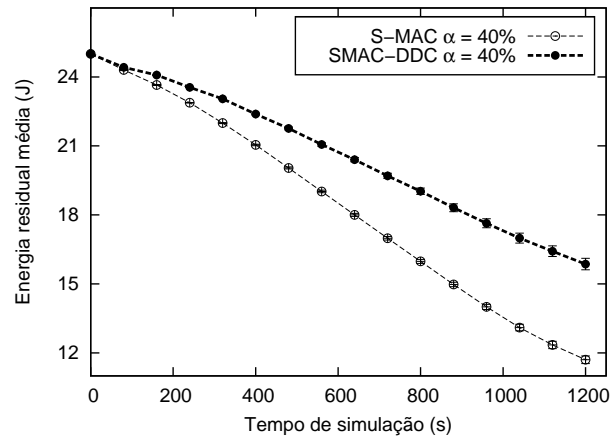
4.5 Consumo de Energia

Nesta seção será analisado o consumo de energia dos protocolos e avaliado o ganho alcançado com a abordagem proposta. A figura 4.7 apresenta os gráficos que comparam o consumo de energia para um cenário de eventos fixos com o valor de $\sigma = 10$. Percebe-se que para ambos os valores de α o protocolo SMAC-DDC reduz o consumo de energia em 30%. Essa redução é alcançada devido à adaptabilidade proposta pelo SMAC-DDC. Nesse protocolo, os nós não pertencentes à rota têm o seu *duty cycle* reduzido para $\alpha/2$, o que abrevia o tempo que o nó permanece com o seu rádio ligado e diminui o consumo de energia. Nota-se também que apesar do SMAC-DDC aumentar o *duty cycle* dos nós pertencentes à rota, o consumo médio de energia da rede não é acrescido, pois o *duty cycle* dos nós é alterado somente durante o sensoriamento do evento. Após o término do mesmo, o *duty cycle* tem o seu valor retornado ao limite inicial de $\alpha/2$.

Em contraste, o protocolo S-MAC adota um valor igual de *duty cycle* para todos os nós, por isso, esse protocolo apresenta um consumo maior de energia quando comparado ao SMAC-DDC. Enquanto no protocolo SMAC o valor *duty cycle* no início é α , no protocolo SMAC-DDC o valor é $\alpha/2$. Além disso, no protocolo SMAC esse valor é pré-definido e permanece inalterado durante todo o tempo, não se adaptando ao fluxo de dados da rede.

Analisando também o consumo de energia médio da rede em diferentes cenários de eventos, observa-se que o comportamento do protocolo SMAC-DDC é melhor do que o protocolo S-MAC. No gráfico da figura 4.8-(a) são mostrados os resultados para um cenário de eventos aleatórios e pode-se notar que o protocolo SMAC-DDC apresenta um consumo de energia 20% menor que o protocolo S-MAC. Essa redução no consumo de energia é alcançada pela utilização de um valor de *duty cycle* menor para os nós que não estão na rota, diminuindo o tempo que o rádio fica ligado. Também se observa que uma quantidade maior de eventos na rede aumenta o consumo de energia. Fato esse explicado, pois, uma quantidade maior de dados será trafegada na rede implicando em um número maior de transmissões realizadas.

Avaliando o consumo de energia em um cenário de eventos fixos, na figura 4.8-(b), observa-se que o protocolo SMAC-DDC diminui em 20% o consumo de energia. Esse

(a) $\alpha = 30\%$.(b) $\alpha = 40\%$.**Figura 4.7.** Comparação do consumo de energia para diferentes valores de α .

comportamento é esperado uma vez que, com esse protocolo os nós trabalham com dois valores de *duty cycle*; o valor menor é usado quando não faz parte de nenhuma rota e o maior quando o nó pertence à rota para escoamento de dados em direção ao nó *sink*. Contudo nesse cenário de eventos fixos, a distância entre o nó que sensoria o evento e o nó de monitoramento é sempre maior que a distância mínima pré-definida explicando o fato de que apesar de ocorrerem poucos eventos a rede apresenta um consumo de energia elevado para os dois protocolos.

4.6 Taxa de Entrega

Nesta seção será apresentado os resultados para última métrica escolhida para avaliação: Taxa de Entrega. Essa métrica contabiliza todos os pacotes que chegaram ao nó *sink* durante todo o tempo de simulação. A figura 4.9 mostra os resultados para os dois

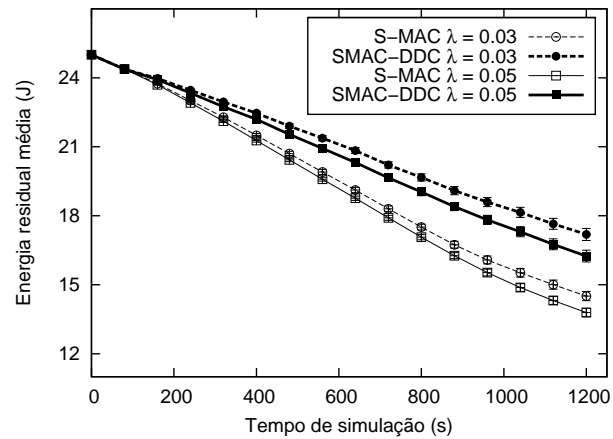
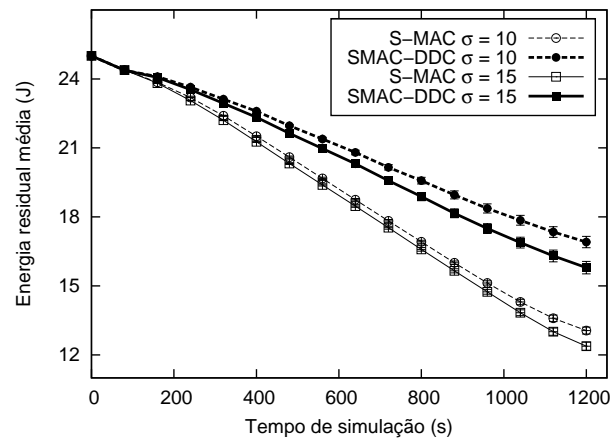
(a) Diferentes valores de λ para eventos aleatórios.(b) Diferentes valores de σ para eventos fixos.

Figura 4.8. Comparação do consumo de energia para diferentes formas de evento.

protocolos com a variação do α em um cenário de eventos fixos com o valor de $\sigma = 10$. Nota-se que com α no valor 30% os dois protocolos entregam praticamente a mesma quantidade de dados. A esses resultados foi aplicado o teste da diferença das médias descrito em [Downie & Heath, 1965] e conclui-se que os dois protocolos demonstram desempenho similar na entrega de dados para esse valor de α . Percebe-se que para esse valor de α , as razões utilizadas pelo protocolo SMAC-DDC ($\alpha/2$ e 2α) favorecem o seu desempenho quando comparado ao protocolo S-MAC.

Em contrapartida, com o α de 40% verifica-se que o S-MAC entrega 22% de dados a mais que o SMAC-DDC. Essa diferença é explicada pelo fato de que o protocolo S-MAC com o valor α maior viabiliza uma fase de descoberta e criação das rotas mais rápida proporcionando um maior fluxo de dados na rede. Enquanto que, nessa mesma fase, o protocolo SMAC-DDC trabalha com um valor de *duty cycle* configurado para

$\alpha/2$, retardando esse processo de descoberta de rotas. Somente depois dessa fase o protocolo SMAC-DDC altera o valor do *duty cycle* para 2α dos nós pertencentes à rota.

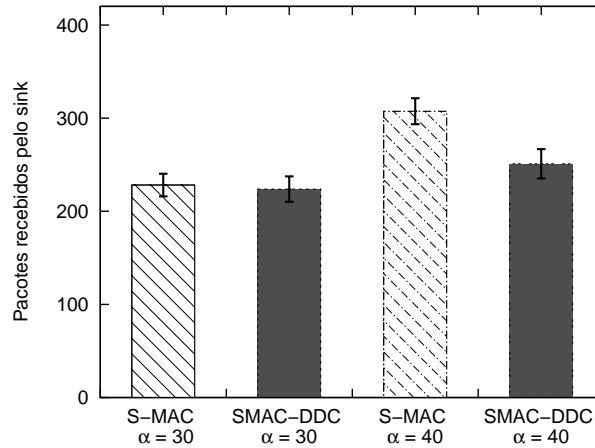


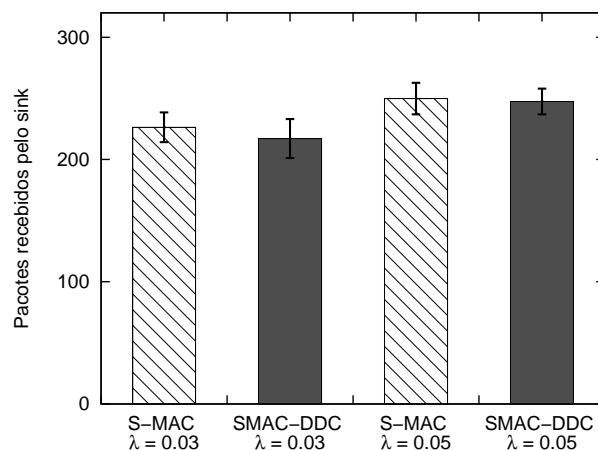
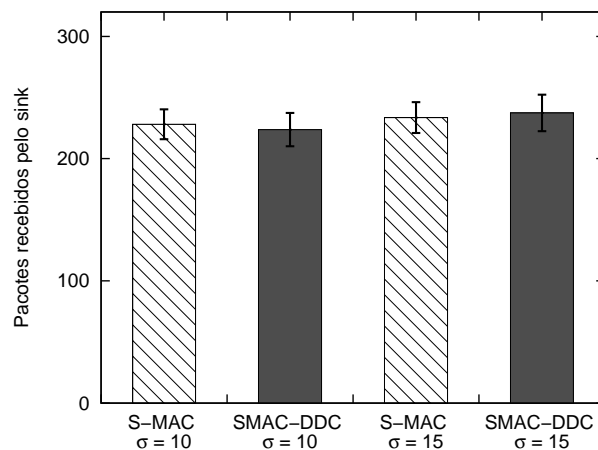
Figura 4.9. Taxa de entrega de dados para diferentes valores de α .

A figura 4.10 exibe os resultados para diferentes tipos de eventos. Para essa comparação o valor de α escolhido foi 30%. Na figura 4.10-(a) tem-se os resultados para eventos aleatórios. Pode-se concluir que os dois protocolos, quando sujeitos ao mesmo tráfego apresentam resultados bem semelhantes. Esse resultado é observado porque o SMAC-DDC apesar de trabalhar inicialmente com o valor de *duty cycle* menor, ao sensoriar um evento esse valor é aumentado para os nós que estão no caminho entre o evento e o nó *sink*. Essa característica propicia um aumento no volume de dados que chegam ao *sink*, aproximando o desempenho desse protocolo ao protocolo S-MAC.

A figura 4.10-(b) apresenta os resultados para um cenário de eventos fixos com a variação do valor de σ . Observa-se o mesmo comportamento no cenário de eventos aleatórios, ambos os protocolos apresentaram desempenho semelhantes diante da variação da quantidade de eventos. É importante evidenciar que o aumento da quantidade de eventos implica em um aumento na quantidade de dados entregues ao nó *sink*, como se observa no gráfico da figura 4.10.

4.7 Conclusão

Neste capítulo foi feita uma avaliação dos fatores que impactam a latência dos pacotes que chegam ao nó *sink*. Verifica-se que o parâmetro *duty cycle* exerce uma influência expressiva sobre o valor da latência, pois esse parâmetro determina o tempo que o nó sensor irá permanecer com o seu rádio ligado.

(a) Diferentes valores de λ para eventos aleatórios.(b) Diferentes valores de σ para eventos fixos.**Figura 4.10.** Comparação da taxa de entrega para diferentes formas de evento.

Também foi avaliado o desempenho do protocolo proposto SMAC-DDC frente ao protocolo S-MAC. Observa-se que em todos os cenários escolhidos o protocolo SMAC-DDC reduziu consideravelmente a latência dos pacotes e também proporcionou uma boa economia de energia. Esse ganho é propiciado pela adaptação do *duty cycle* de acordo com o fluxo de dados. Entretanto, o protocolo proposto manteve a taxa de entrega do S-MAC nos cenários escolhidos quando o valor de α é 30. Testes estatísticos foram aplicados aos resultados e concluiu-se que esses protocolos possuem desempenho semelhante na coleta de dados. Todavia, quando o valor de α é 40% esse mesmo comportamento não foi observado.

CONCLUSÕES E DIREÇÕES FUTURAS

Neste trabalho foi proposto um novo protocolo MAC que adapta a duração do *duty cycle* do nó sensor de acordo com o fluxo de dados na rede. Essa adaptação foi conseguida por intermédio de um projeto integrado de camadas que rompe os padrões tradicionais de comunicação entre camadas e permite que a camada de rede compartilhe informações com a camada de enlace. O objetivo desse protocolo é, com a adaptação da duração do *duty cycle*, reduzir a latência e minimizar o consumo de energia na rede sem interferir na taxa de entrega de dados. Resultados de simulação mostraram que a duração do período do *duty cycle* possui grande interferência na latência dos pacotes que chegam ao nó *sink*. Além disso, foi possível verificar que a dinamicidade do *duty cycle* proporciona uma melhora na coleta de dados, reduzindo significativamente a latência sem alterar a taxa de entrega. Outro ponto importante é que o protocolo SMAC-DDC se mostrou eficiente no consumo de energia, reduzindo o mesmo quando comparado ao protocolo S-MAC.

Como trabalhos futuros, planeja-se estudar outras maneiras para construir um projeto integrado, além do compartilhamento de diferentes informações entre as camadas. Pretende-se também estudar mais a influência que o *duty cycle* exerce no funcionamento da rede. Pretende-se também aplicar a adaptação do *duty cycle* em outras formas de comunicação como disseminação de dados e cooperação. Outro ponto importante a ser analisado é a influência do tráfego de dados na duração do *duty cycle* e assim estudar uma forma de interligar esses valores e assim ajustar o *duty cycle* em função da quantidade de tráfego. Outra vertente a ser estudada é a criação de protocolos mais específicos para RSSFs, considerando as características peculiares desse tipo de rede com a finalidade de alcançar um melhor desempenho da rede e melhorar o aproveitamento de todos os recursos disponíveis.

REFERÊNCIAS BIBLIOGRÁFICAS

- Akyildiz, I.; Su, W.; Sankarasubramaniam, Y. & Cayirci, E. (2002). A survey on sensor networks. *Communications Magazine, IEEE*, 40(8):102–114.
- Al-Karaki, J. & Kamal, A. (2004). Routing techniques in wireless sensor networks: a survey. *Wireless Communications, IEEE*, 11(6):6–28.
- Borgia, E.; Conti, M. & Delmastro, F. (2006). Mobileman: design, integration, and experimentation of cross-layer mobile multihop ad hoc networks. In *Communications Magazine, IEEE*, pp. 80–85.
- Chen, F. (September 2005). Simulation of wireless sensor nodes using s-mac. Master's thesis, University of Erlangen-Nuremberg, Department of Computer Science 7.
- Correia, L. H. A.; Macedo, D. F.; dos Santos, A. L.; Nogueira, J. M. S. & Loureiro, A. A. F. (2005). Uma taxonomia para protocolos de controle de acesso ao meio em redes de sensores sem fio. Technical report, Universidade Federal de Minas Gerais, Departamento de Ciência da Computação.
- Crossbow (2008). Mica2 868, 916 mhz. <http://www.xbow.com/>.
- Demirkol, I.; Ersoy, C. & Alagoz, F. (2006). Mac protocols for wireless sensor networks: a survey. *Communications Magazine, IEEE*, 44(4):115--121.
- Ding, J.; Sivalingam, K.; Kashyapa, R. & Chuan, L. J. (6-9 Oct. 2003). A multi-layered architecture and protocols for large-scale wireless sensor networks. *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, 3:1443–1447.
- Downie, N. M. & Heath, R. W. (1965). *Basic Statistical Methods*. Harper and Row.
- Enz, C.; El-Hoiydi, A.; Decotignie, J.-D. & Peiris, V. (2004). Wisenet: an ultralow-power wireless sensor network solution. *Computer*, 37(8):62–70.

- Fang, Y. & McDonald, A. (4-7 Oct. 2004). Dynamic codeword routing (dcr): a cross-layer approach for performance enhancement of general multi-hop wireless routing. *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pp. 255–263.
- Goldsmith, A. & Wicker, S. (2002). Design challenges for energy-constrained ad hoc wireless networks. *Wireless Communications, IEEE*, 9(4):8–27.
- Heidemann, J. & Silva, F. (2003). Matching data dissemination algorithms to application requirements. In *In Proceedings of the ACM SenSys Conference*, pp. 218--229. ACM Press.
- Heinzelman, W. B.; Chandrakasan, A. P. & Balakrishnan, H. (2000). Energy-efficient communication protocol for wireless microsensor networks. In *Hawaii International Conference on System Sciences*, volume 8, Maui, Hawaii.
- Heinzelman, W. B.; Chandrakasan, A. P. & Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. In *IEEE Transactions on Wireless Communications*, volume 4, pp. 660 – 670.
- Intanagonwiwat, C.; Govindan, R. & Estrin, D. (2000). Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pp. 56--67, New York, NY, USA. ACM Press.
- Jain, R. (1991). *The Art of Computer Systems Performance Analysis: techniques for experimental design, measurement, simulation and modeling*. John Wiley.
- Jurdak, R. (2007). *Wireless Ad Hoc and Sensor Networks: A Cross-Layer Design Perspective*. Springer-Verlag.
- Kalidindi, R.; Ray, L.; Kannan, R. & Iyengar, S. S. (junho 2003). Distributed energy-aware mac protocol for wireless sensor networks. *International Conference on Wireless Networks, Las Vegas, Nevada*.
- Kawadia, V. & Kumar, P. (2005). A cautionary perspective on cross-layer design. *Wireless Communications, IEEE*, 12(1):3–11.
- Kulik, J.; Heinzelman, W. & Balakrishnan, H. (2002). Negotiation-based protocols for disseminating information in wireless sensor networks. *Wirel. Netw.*, 8(2/3):169--185.

- Lin, R.; Wang, Z.; Li, Y. & Sun, Y. (2005). A scalable energy efficient medium access control protocol for wireless sensor networks. *ICIC - International Construction Innovations Conference*.
- Lindsey, S. & Raghavendra, C. S. (2002). Pegasus: Power-efficient gathering in sensor information systems. In *Aerospace Conference Proceedings*, volume 3.
- Machado, M.; Goussevskaia, O.; Mini, R.; Rezende, C.; Loureiro, A.; Mateus, G. & Nogueira, J. (2005). Data dissemination in autonomic wireless sensor networks. *Selected Areas in Communications, IEEE Journal on*, 23(12):2305–2319.
- ns2 (2002). The network simulator. www.isi.edu/nsnam/ns.
- Polastre, J.; Hill, J. & Culler, D. (2004). Versatile low power media access for wireless sensor networks. *Sensys*.
- Pottie, G. & Kaiser, W. (2000). Embedding the internet wireless integrated network sensors. *Communications of the ACM*, 43:51–58.
- Puccinelli, D. & Haenggi, M. (2005). Wireless sensor networks: applications and challenges of ubiquitous sensing. *Circuits and Systems Magazine, IEEE*, 5(3):19–31.
- Rajendran, V.; Obraczka, K. & Garcia-Luna-Aceves, J. J. (2006). Energy-efficient, collision-free medium access control for wireless sensor networks. *Wirel. Netw.*, 12(1):63–78.
- Ross, S. M. (1996). *Simulation (Statistical Modeling and Decision Science)*. Academic Press, second edição.
- Ruiz, L. B.; Correia, L. H. A.; Vieira, L. F. M.; Macedo, D. F.; Nakamura, E. F.; Figueiredo, C. M. S.; Vieira, M. A. M.; Maia, E. H. B.; Camara, D.; Loureiro, A. A. F.; Nogueira, J. M. S.; da Silva Jr., D. C. & Fernandes, A. O. (2004). Arquiteturas para Redes de Sensores Sem Fio. In *22 Simpósio Brasileiro de Redes de Computadores*, pp. 167 -- 218.
- Sohrabi, K.; Gao, J.; Ailawadhi, V. & Pottie, G. (2000). Protocols for self-organization of a wireless sensor network. *Personal Communications, IEEE*, 7(5):16–27.
- Souto, E.; Aschoff, R.; Sadok, D. H. & Kelner, J. (2007). Projeto cross-layer entre as camadas mac, roteamento e aplicação em redes de sensores. In *Anais do XXV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, volume 1, pp. 45–58, Pará, Belém.

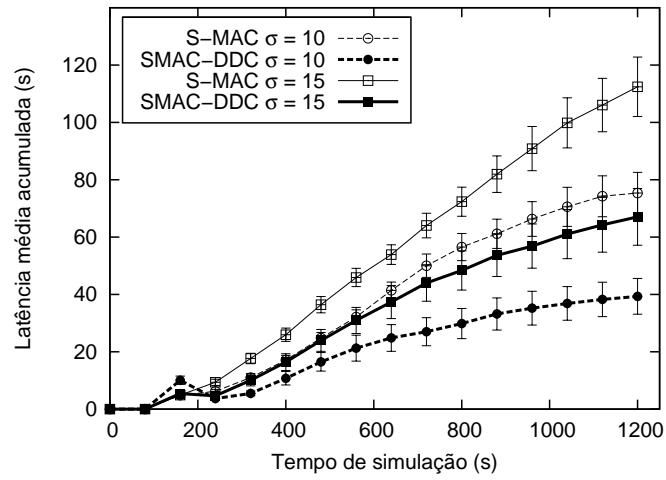
- Srivastava, V. & Motani, M. (2005). Cross-layer design: a survey and the road ahead. *Communications Magazine, IEEE*, 43(12):112--119.
- Tanenbaum, A. S. (2003). *Redes de Computadores*. Elsevier, Rio de Janeiro, trad. 4 ed. edição.
- Van Dam, T. & Langendoen, K. (2003). An adaptive energy-efficient mac protocol for wireless sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pp. 171--180, New York, NY, USA. ACM Press.
- Van Hoesel, L.; Nieberg, T.; Wu, J. & Havinga, P. (Dec. 2004). Prolonging the lifetime of wireless sensor networks by cross-layer interaction. *Wireless Communications, IEEE [see also IEEE Personal Communications]*, 11(6):78--86.
- Wang, L.-C. & Wang, C.-W. (21-23 March 2004). A cross-layer design of clustering architecture for wireless sensor networks. *Networking, Sensing and Control, 2004 IEEE International Conference on*, 1:547--552 Vol.1.
- Ye, W. & Heidemann, J. (2003). Medium access control in wireless sensor networks. Technical report, University of Southern California, Information Sciences Institute.
- Ye, W.; Heidemann, J. & Estrin, D. (2002). An energy-efficient mac protocol for wireless sensor networks. *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 3:1567--1576 vol.3.
- Ye, W.; Heidemann, J. & Estrin, D. (2004). Medium access control with coordinated adaptive sleeping for wireless sensor networks. *Networking, IEEE/ACM Transactions on*, 12(3):493--506.

VARIAÇÃO DA DISTÂNCIA MÍNIMA DOS EVENTOS ESTÁTICOS

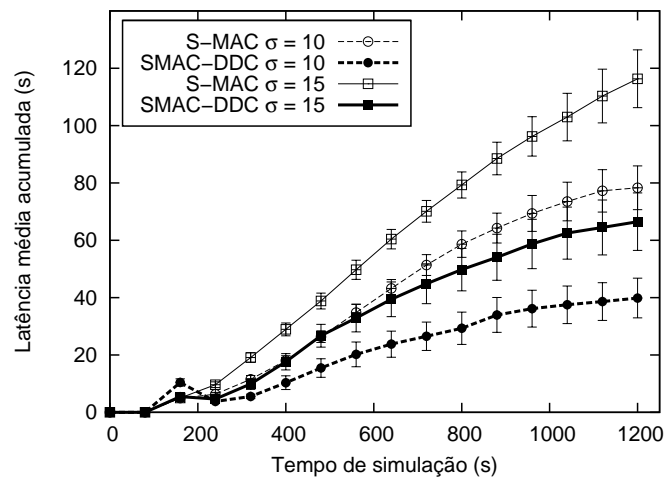
Neste apêndice, avalia-se a influência que o valor da distância mínima dos eventos estáticos exerce sobre a latência, consumo de energia e taxa de entrega dos pacotes ao nó *sink*.

A.1 Latência

Nesta seção é apresentado os resultados para a latência com a variação da distância mínima dos eventos ao nó *sink*. Os gráficos apresentados pelas figuras A.1-(a), A.1-(b), A.2-(a) e A.2-(b) mostram o valor da latência quando a distância mínima assumo o valor de 20, 40, 60 e 80 metros. Nesses resultados o valor utilizado para $\alpha = 30\%$.

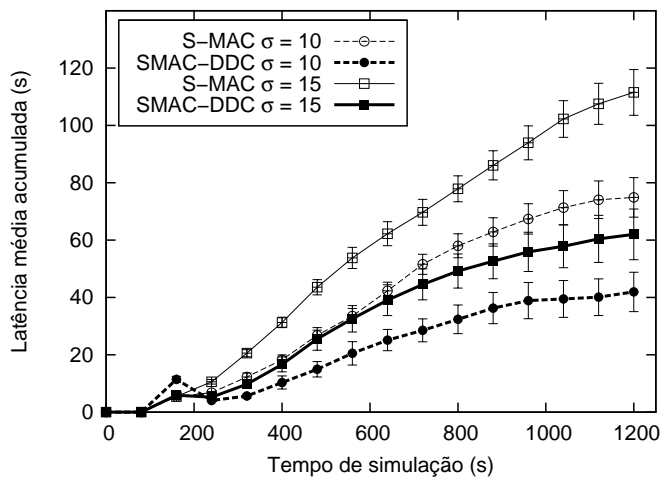


(a) Distância mínima = 20 m.

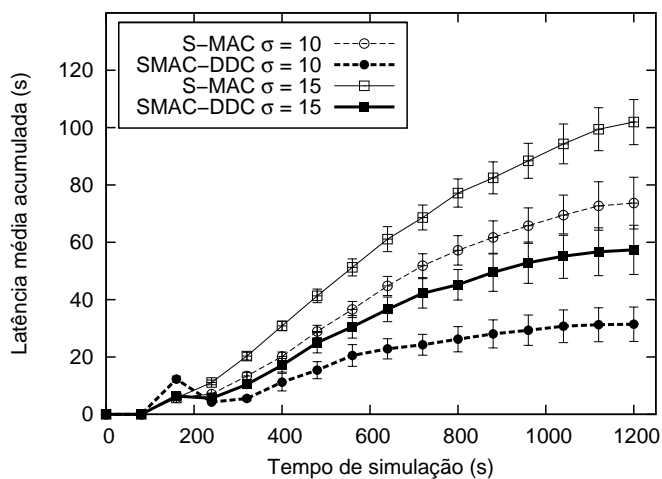


(b) Distância mínima = 40 m.

Figura A.1. Latência média acumulada para distância mínima igual a 20 e 40 metros.



(a) Distância mínima = 60 m.

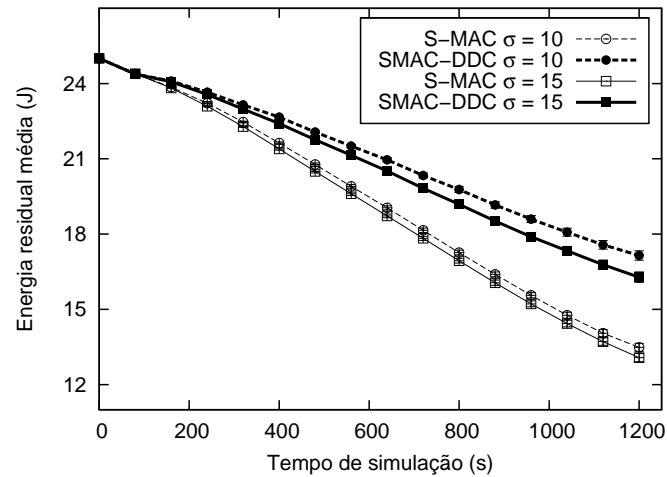


(b) Distância mínima = 80 m.

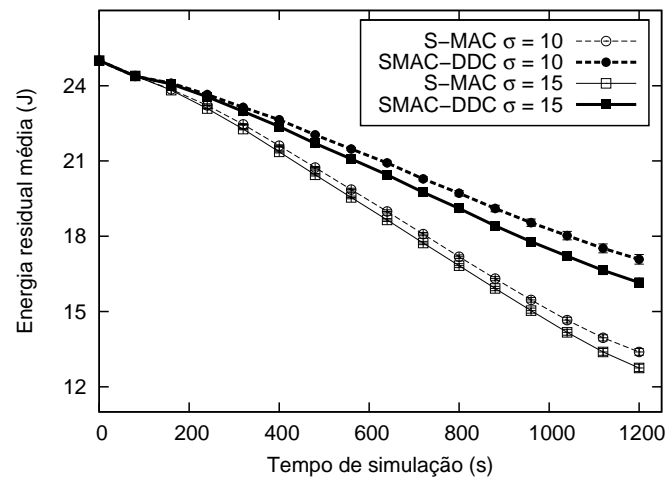
Figura A.2. Latência média acumulada para distância mínima igual a 60 e 80 metros.

A.2 Consumo de Energia

Esta seção apresenta uma comparação do consumo médio de energia quando o valor da distância mínima dos eventos fixos é variado. As figuras A.3-(a), A.3-(b), A.4-(a) e A.4-(b) apresentam a energia residual média quando a distância mínima é 20, 40, 60 e 80 metros.

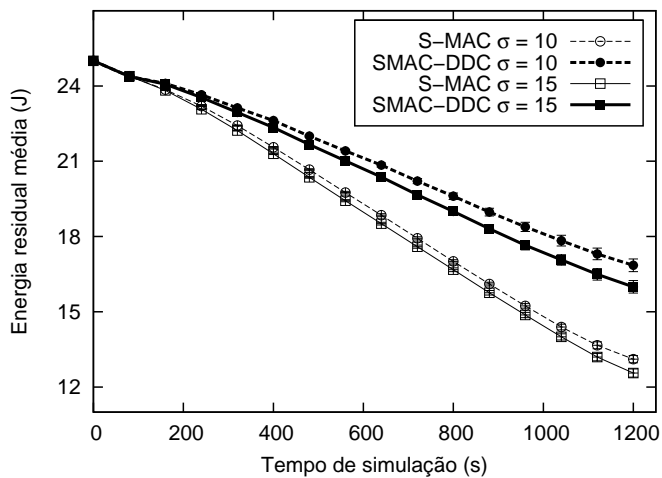


(a) Distância mínima = 20 m.

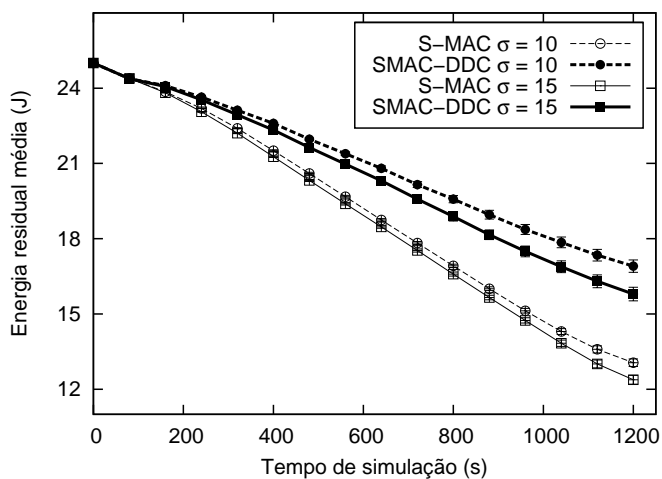


(b) Distância mínima = 40 m.

Figura A.3. Energia residual média para distância mínima igual a 20 e 40 metros.



(a) Distância mínima = 60 m.

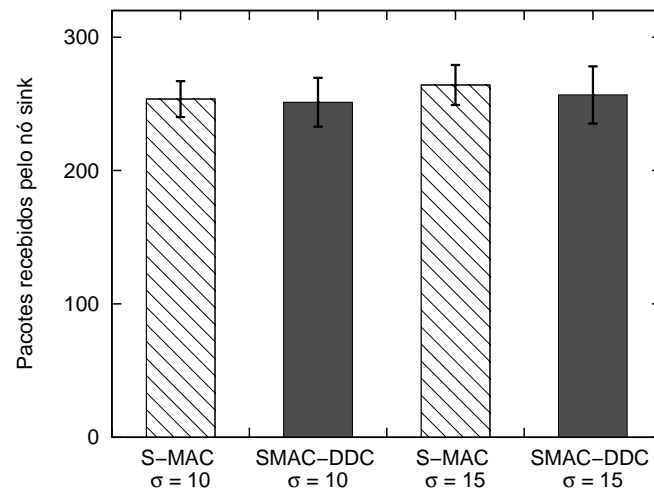


(b) Distância mínima = 80 m.

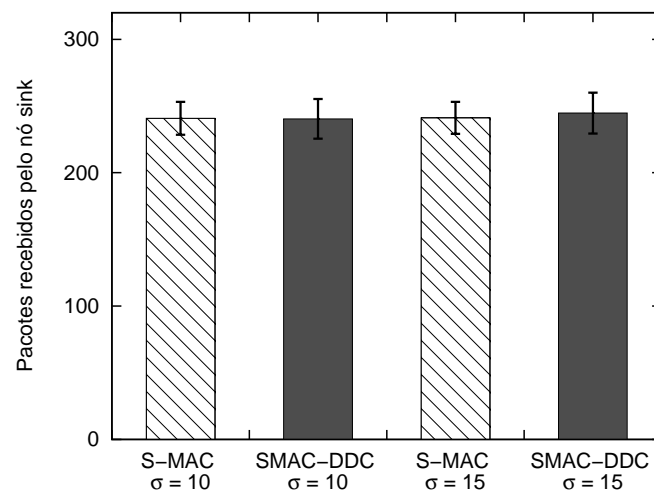
Figura A.4. Energia residual média para distância mínima igual a 60 e 80 metros.

A.3 Taxa de Entrega

Nesta seção serão apresentados os resultados para a métrica taxa de entrega quando o valor da distância mínima dos eventos fixos é variado. As figuras A.5-(a), A.5-(b), A.6-(a) e A.6-(b) apresentam, respectivamente, a variação da distância mínima para 20, 40, 60 e 80 metros.

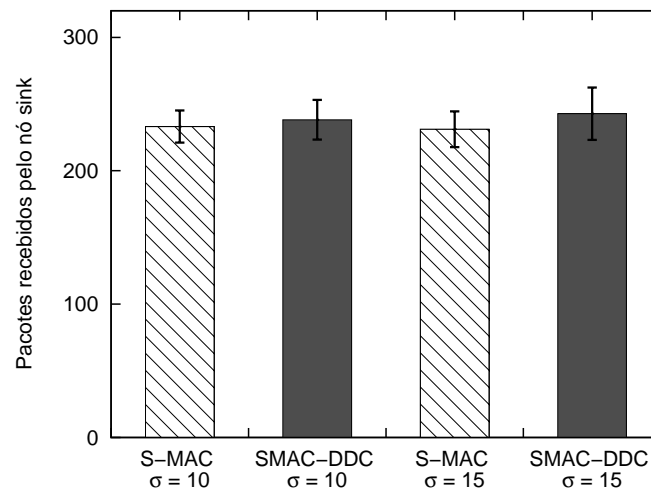


(a) Distância mínima = 20 m.

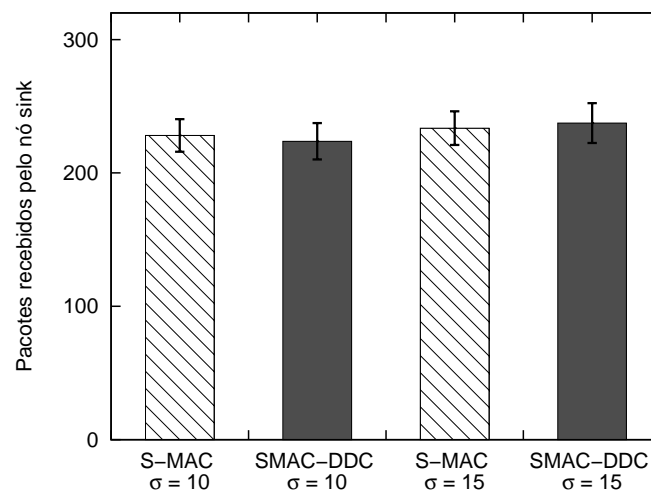


(b) Distância mínima = 40 m.

Figura A.5. Taxa de entrega para distância mínima igual a 20 e 40 metros.



(a) Distância mínima = 60 m.



(b) Distância mínima = 80 m.

Figura A.6. Taxa de entrega para distância mínima igual a 60 e 80 metros.

PROJETO EXPERIMENTAL FATORIAL

O objetivo para utilizar um projeto experimental é obter o máximo de informações realizando o mínimo de experimentos possível. Esse procedimento economiza trabalho e uma correta análise do projeto é possível separar os efeitos de vários fatores que podem afetar no desempenho do sistema. Também, é possível determinar se um fator tem um efeito significativo ou se a diferença observada nos resultados é devido a erros de medias ou parâmetros que não foram controlados.

Neste apêndice será descrito, passo a passo, como um projeto experimental $2^k r$ apresentado na seção 4.2 é calculado. Essa descrição é feita baseada no livro do Rai Jain [Jain, 1991]. A seção B.1 descreve os principais termos usados na análise de um projeto experimental e a seção B.2 apresenta, passo-a-passo, como foram efetuados os cálculos para o projeto Experimental $2^3 r$ apresentado no texto.

B.1 Terminologia

Esta seção define os termos que são freqüentemente usados em um projeto experimental na concepção e análise dos resultados. São eles:

Variável de Resposta: O resultado de um experimento é chamado de variável de resposta. Geralmente a variável de resposta é a medida do desempenho do sistema.

Fatores: Cada variável que afeta a variável de resposta e tem diversas alternativas é chamado de fator. Podem ser conhecidos também como variáveis preditoras ou parâmetros.

Níveis: O valor que cada fator pode assumir é chamado de nível. Em outras palavras, um nível constitui de uma alternativa de valor para um fator.

Fatores Primários: são aqueles fatores cujos os efeitos necessitam ser quantificados e através do projeto experimental pode-se quantificar esses efeitos.

Fatores Secundários: são fatores que influenciam no desempenho mas não há interesse na avaliação do impacto dos mesmos.

Replicação: A repetição de alguns ou todos os experimentos é chamada de replicação. Por exemplo, se todos os experimentos no estudo de caso foram repetidos 3 vezes, então é dito que o estudo de caso teve 3 replicações.

Projeto: Um projeto experimental consiste da especificação do número de experimentos, combinações para os níveis de fatores, e o número de replicações para cada experimento.

Interação: A interação entre dois fatores A e B existe quando um fator depende do nível do outro.

B.2 Projeto Experimental $2^k r$

Um projeto experimental é usado para determinar o efeito de k fatores, cada um tendo duas alternativas ou níveis. Essa é uma classe de projeto fatorial que apresenta uma fácil análise, pois, os fatores apresentam apenas dois níveis de variação. Entretanto um projeto experimental 2^k é possível apresentar erros experimentais por não haver repetição dos experimentos. Podem-se quantificar esses erros com a repetição de um experimento sob uma mesma configuração de níveis-fatores. Então, cada um dos 2^k experimentos serão repetidos r vezes, o que gera $2^k r$ observações. Um projeto experimental que apresenta repetições é conhecido como $2^k r$.

Nesta seção é apresentado todo o cálculo para efetuar um projeto experimental $2^k r$ utilizando como exemplo o projeto experimental $2^3 r$ apresentado na seção 4.2, no qual é avaliado o impacto de 3 fatores no valor da latência, cada fator com dois níveis de variação. Para esse projeto foram efetuadas 33 repetições com a finalidade de aumentar a confiança dos dados finais obtidos.

Para o projeto foram definidas três variáveis x_σ , x_R e x_D para representar os parâmetros, e os seus respectivos níveis ficaram definidos da seguinte forma:

$$x_\sigma = \begin{cases} -1 & \text{se a quantidade de eventos for 20} \\ 1 & \text{se a quantidade de eventos for 10} \end{cases}$$

$$x_R = \begin{cases} -1 & \text{se o raio de comunicação for 15} \\ 1 & \text{se o raio de comunicação for 25} \end{cases}$$

$$x_D = \begin{cases} -1 & \text{se o } \textit{duty cyle} \text{ for 10\%} \\ 1 & \text{se o } \textit{duty cyle} \text{ for 40\%} \end{cases}$$

O valor da *Latência* em segundos pode agora ser regredido em x_R , x_D e x_σ usando uma expressão não-linear do modelo de regressão da seguinte forma:

$$\begin{aligned} \textit{Latência} = & q_0 + q_\sigma x_\sigma + q_R x_R + q_D x_D + q_{\sigma R} x_\sigma x_R + \\ & q_{\sigma D} x_\sigma x_D + q_{RD} x_R x_D + q_{\sigma RD} x_\sigma x_R x_D + e \end{aligned}$$

Após a execução dos experimentos e sumarização dos dados a média aritmética para cada configuração dos parâmetros é apresentada na tabela B.1. O próximo passo é calcular o efeito de cada parâmetro no valor da latência. Para isso, basta multiplicar as entradas na coluna do parâmetro pela coluna que contém a média e colocar a soma embaixo da coluna. Essa soma deve ser efetuada para cada parâmetro, inclusive as colunas que representam as interações entre eles. A soma de cada coluna é dividida pela quantidade de experimentos, neste caso 8, para obtenção dos coeficientes correspondentes ao modelo de regressão. A tabela B.1 apresenta os cálculos para cada efeito nas últimas linhas.

<i>I(média)</i>	x_σ	x_R	x_D	$x_\sigma x_R$	$x_\sigma x_D$	$x_R x_D$	$x_\sigma x_R x_D$	latência
1	-1	-1	-1	1	1	1	-1	154,82
1	-1	-1	1	1	-1	-1	1	94,72
1	-1	1	-1	-1	1	-1	1	202,02
1	-1	1	1	-1	-1	1	-1	102,83
1	1	-1	-1	-1	-1	1	1	116,63
1	1	-1	1	-1	1	-1	-1	45,82
1	1	1	-1	1	-1	-1	-1	130,33
1	1	1	1	1	1	1	1	54,39
901,55	-207,22	77,57	-306,05	-33,04	12,54	-44,21	33,97	<i>Total</i>
112,69	-25,90	9,70	-38,26	-4,13	1,57	-5,53	4,25	<i>Total/8</i>

Tabela B.1. Resultados do Projeto Experimental 2^3r .

A importância de um parâmetro é medida pela proporção da resposta que é explicada pelo parâmetro. Assim, para efetuar esse cálculo, é necessário primeiramente determinar a soma total dos quadrados (SST) dado pela seguinte equação:

$$\text{Total da variação da Latência} = \text{SST} = \sum_{i,j} (y_{ij} - \bar{y}_{..})^2$$

na qual y_{ij} representa o valor da latência em uma replicação para uma determinada configuração e \bar{y} representa o valor médio da latência para todas as replicações em todos os experimentos. O SST obtido representa a soma da diferença dos valores observados pelo valor predito.

Para um projeto 2^3 , o SST pode ser dividido em oito partes como segue:

$$\sum_{i,j} (y_{ij} - \bar{y}_{..})^2 = 2^3 r q_\sigma^2 + 2^3 r q_R^2 + 2^3 r q_D^2 + 2^3 r q_{\sigma R}^2 + 2^3 r q_{\sigma D}^2 + 2^3 r q_{RD}^2 + 2^3 r q_{\sigma RD}^2 + \sum_{i,j} e_{ij}^2$$

$$\text{SST} = \text{SS}\sigma + \text{SS}R + \text{SS}D + \text{SS}\sigma R + \text{SS}\sigma D + \text{SS}RD + \text{SS}\sigma RD + \text{SSE}$$

na qual cada SS(soma dos quadrados) corresponde à expressão acima. Assim, $\text{SS}\sigma$, $\text{SS}R$, $\text{SS}D$, $\text{SS}\sigma R$, $\text{SS}\sigma D$, $\text{SS}RD$ e $\text{SS}\sigma RD$ são as variações explicadas pelos parâmetros σ , R , D , e as interações σR , σD , RD e σRD respectivamente. O SSE explica a variação atribuída aos erros experimentais.

Então, a proporção explicada por cada parâmetro é expressa como uma fração, por exemplo:

$$\text{Fração explicada pelo parâmetro } \sigma = \frac{\text{SS}\sigma}{\text{SST}}$$

segundo a avaliação do projeto, a seguir é apresentado o cálculo para todas as somas

dos quadrados e suas respectivas proporções.

$$\begin{array}{ll}
 SS\sigma \implies 177.123,90 & SSRD \implies 8.062,42 \\
 SSR \implies 24.823,54 & SS\sigma RD \implies 4.761,22 \\
 SSD \implies 386.369,65 & SST \implies 691.604,45 \\
 SS\sigma R \implies 4.502,88 & SSE \implies 85.311,78 \\
 SS\sigma D \implies 649,02 &
 \end{array}$$

Efetuada a divisão das somas dos quadrados (SS) pela soma total dos quadrados (SST) verifica-se que os parâmetros σ , R e D impactam, respectivamente, em 25%, 3% e 55% no valor da latência e as interações entre eles impactam em menos de 2%. Os erros experimentais explicam cerca de 12,8% do valor da latência.

Assim, a equação final para o modelo de regressão do cálculo da latência fica definido da seguinte forma:

$$\begin{aligned}
 \widehat{\text{Latência}} = & 112,69 - 25,90x_\sigma + 9,70x_R - 38,36x_D - 4,13x_\sigma x_R + \\
 & 1,57x_\sigma x_D - 5,53x_R x_D + 4,25x_\sigma x_R x_D
 \end{aligned}$$

Parâmetro	Efeito	Porcentagem de Explicação	Intervalo de Confiança
I	112.69		(112.57, 112.80)
σ	-25.90	25.6	(-26.01, -25.78)
R	9.70	3.5	(9.58, 9.81)
D	-38.26	55.8	(-38.37, -38.14)
σR	-4.13	0.6	(-4.24, -4.01)
σD	1.57	0.1	(1.45, 1.68)
RD	-5.53	1.0	(-5.64, -5.41)
σRD	4.25	0.6	(4.13, 4.36)
e		12.8	