UNIVERSIDADE FEDERAL RURAL DO SEMIÁRIDO
UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

WELLIANA BENEVIDES RAMALHO

UM ALGORITMO PARA FORMAÇÃO DE CONJUNTO DE SENSORES EM REDES DE SENSORES SEM FIO UTILIZANDO RELACIONAMENTO CRUZADO ENTRE CAMADAS

MOSSORÓ – RN
2011

WELLIANA BENEVIDES RAMALHO

UM ALGORITMO PARA FORMAÇÃO DE CONJUNTO DE SENSORES EM REDES DE SENSORES SEM FIO UTILIZANDO RELACIONAMENTO CRUZADO ENTRE CAMADAS

Dissertação apresentada ao Mestrado de Ciência da Computação – associação ampla entre a Universidade do Estado do Rio Grande do Norte e a Universidade Federal Rural do Semiárido, para a obtenção do título de Mestre em Ciência da Computação.

Orientadores: Prof. Iguatemi Eduardo da Fonseca

Prof. Marcelo Sampaio de Alencar

MOSSORÓ – RN
2011

Welliana Benevides Ramalho

UM ALGORITMO PARA FORMAÇÃO DE CONJUNTO DE SENSORES EM REDES DE SENSORES SEM FIO UTILIZANDO RELACIONAMENTO CRUZADO ENTRE CAMADAS

Dissertação apresentada ao Programa de Pós-graduação em Ciência da Computação da UERN-UFERSA, como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Aprovado em 08 de Agosto de 2011

BANCA EXAMINADORA

_____
Prof. Iguatemi Eduardo da Fonseca
D.Sc.

_____
Prof. Marcelo Sampaio de Alencar
Ph.D.

_____
Prof. Hélcio Wagner da Silva
D.Sc.

_____
Prof. Waslon Terllizzie Araújo Lopes
D.Sc.

Dedico este trabalho aos meus pais José e Luíza, meu exemplo de força, coragem e determinação.

AGRADECIMENTOS

A Deus por está presente em minha vida, por Nele encontrar a paz nos momentos difíceis.

Ao meu professor e orientador Iguatemi Eduardo da Fonseca, pelos bons conselhos e por ter me conduzido neste trabalho. Ao professor Marcelo Sampaio de Alencar, também meu orientador, pelas colocações e conselhos de suma importância.

Aos meus pais, José e Luíza pelo amor, carinho, educação e exemplo de vida que me proporcionaram.

Às minhas irmãs Welia e Whenia pela presença em todos os momentos deste mestrado me incentivando e me animando.

A todas as pessoas de minha família que me incentivaram e torceram por mim, em especial aos meus avós paternos Pedro Ramalho e Joana Ramalho.

A todos os professores do mestrado, em especial aos professores Pedro Fernandes e Marcelino Pereira por estarem presentes desde a graduação, e por me estimular a ingressar no mestrado.

A meu amigo Edsongley que tanto contribui para esse trabalho.

A Natalyany pelas inúmeras conversas, pelo carinho e por ter me dado a “honra de ser sua amiga”, estando presente em muitos momentos de minha vida.

A Aislânia por todos os momentos de dificuldades e vitórias que passamos desde a graduação até o fim do mestrado, por toda a confiança e amizade.

A meus amigos e companheiros de LASIC Phelipe, Alexsandra e Ticiana, com os

quais dividi além do espaço físico, muitos momentos de companheirismo e amizade.

A Mário, meu amigo e “psicólogo” pelas várias vezes que o incomodei com meus problemas, pelas cortadas de leve, pelas conversas e brincadeiras, pelo incentivo e torcida.

A Cleone pelas inúmeras dicas e conselhos no decorrer do mestrado.

Aos colegas que de alguma forma me ajudaram durante esses dois anos e meio de mestrado, em especial a Marytza, Fernando, Lenardo, Mizael, Tullyo, Leonardo, Luiz Cláudio, Mailson, Sebastião, Gracon, Cláubio, Carlos Fran e Rangel.

Às minhas amigas Fernanda, Missicleide, Ariely e Camila por estarem sempre torcendo pelo meu sucesso.

À Universidade do Estado do Rio Grande do Norte e a Universidade Federal Rural do Semiárido pelo apoio institucional e incentivo.

Agradeço ao CNPq e a Capes pelo auxílio financeiro concedido durante o mestrado.

Enfim, agradeço a todos os funcionários do MCC UERN/UFERSA e também a todas as pessoas que de alguma forma contribuíram para este trabalho.

RESUMO

As redes de sensores sem fio possuem limitações de energia, processamento, armazenamento e potência de transmissão, o que tem impulsionado o desenvolvimento de protocolos específicos para elas. Na literatura tem se destacado o uso de protocolos baseados no relacionamento cruzado entre camadas e também o uso de arquiteturas baseadas na organização hierárquica, em que a rede é dividida em *clusters*. Ambas as soluções, usadas separadamente, têm conseguido uma redução no uso de recursos de tais redes, prolongando o seu tempo de vida útil. Nesta dissertação é proposto um algoritmo de formação de *clusters* que utiliza o relacionamento cruzado entre camadas abrangendo as camadas de enlace, rede e aplicação para redes de sensores sem fio. O protocolo proposto usa o TDMA (*Time Division Multiple Access*), com intervalos de tempo de tamanho variáveis como método para formação e delimitação de *clusters*. Os resultados obtidos sugerem que essa estratégia pode melhorar significativamente o desempenho da rede, reduzindo o atraso fim a fim, o atraso médio e o consumo de energia, ao mesmo tempo em que a sua escalabilidade é favorecida.

Palavras-chave: Redes de Sensores Sem Fio, Relacionamento Cruzado entre Camadas, *Clusters*.

ABSTRACT

Wireless Sensor networks are compound of energy-constrained sensor nodes, which has motivated the development of specific protocols for them. The literature presents the use of protocols based on the cross-layer interaction and also on the use of architectures based on hierarchical organization that divides the network into clusters. Both solutions are used separately to achieve a reduction in the use of resource by such networks, to extend their lifetime. This dissertation proposes a cluster design algorithm that relies on the cross-layer interaction between the Medium Access Control (MAC), Network and Application layers for wireless sensor networks. The protocol uses Time Division Multiple Access (TDMA), with varying time intervals, in conjunction with a routing protocol, in the cluster design process. The results suggest that this strategy can significantly improve the network performance, reducing the end-to-end delay, the average delay and the energy consumption, along with an improvement in scalability.

Keywords: Wireless Sensor Network, Cross-layer Design, Clusters.

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

RSSF – Rede de Sensores Sem Fio

MAC – *Medium Access Control*

TDMA – *Time Division Multiple Access*

MEMS – *Micro Electro-Mecanical Systems*

GPS – *Global Positioning System*

OSI – *Open Systems Interconnection*

SMAC – *Sensor* MAC

DSR – *Dynamic Source Routing*

LEACH – *Low-Energy Adaptative Clustering Hierarchy*

HETCP – *A Hierarchical Energy Efficient Topology Control Protocol*

ACE – *Algorithm for Cluster Establishment*

KCCTC – *K-Connected Cluster Topology Control Protocol*

SUMÁRIO

Capítulo 1

Introdução

1.1 Contextualização

Os recentes avanços nos dispositivos eletromecânicos, na comunicação sem fio e na capacidade de processamento embarcado, possibilitaram o surgimento e o avanço da pesquisa em Redes de Sensores Sem Fio (RSSF). Isso permite a utilização dessas redes nas mais diversas áreas (KARL e WILLIG, 2005; BORGES NETO, 2009; XIONG, 2009).

Considerada uma tecnologia emergente, as RSSFs são constituídas de nós sensores densamente distribuídos em determinado ambiente, cuja função é monitorar ou controlar as condições do ambiente, como por exemplo, temperatura, pressão, níveis de ruído, fluxo de ar, movimento de veículos, entre outros (KARL e WILLIG, 2005; LIANG, 2005).

Porém, muitas das aplicações de RSSFs devem operar sem intervenção humana, o que torna necessário a autonomia e eficiência da rede para cumprir suas tarefas sob diferentes condições de ambiente, além de inserir novos desafios a serem considerados. As RSSFs podem ser utilizadas para monitorar, rastrear, coordenar e processar diversas aplicações, como por exemplo, operações

de recuperação de desastres, mapeamento de biodiversidade, prédios inteligentes, exploração e produção de petróleo e gás, agricultura de precisão, monitoramento de soldados em aplicações militares, medicina, missões espaciais não tripuladas, explorações subaquáticas, dentre outras (LOUREIRO et al., 2003; KARL e WILLIG, 2005; GUIDONI et al., 2006; BORGES NETO, 2009).

Como muitas das aplicações operam sem intervenção humana e os nós sensores usam baterias como fonte de energia, muitas vezes não é possível recarregar ou substituir a bateria do sensor, isso ocasiona uma restrição de energia na RSSF. E pode levar a uma diminuição no tempo de vida útil da rede, além de provocar um baixo poder de processamento, uma baixa capacidade de armazenamento e um pequeno raio de transmissão.

Essa restrição tem motivado a criação ou o aperfeiçoamento de protocolos que minimizam o uso de recursos da rede, visto que protocolos tradicionais não são adequados (ALMEIDA et al, 2010). Porém, a maioria desses protocolos tem em comum o uso da tradicional arquitetura em camadas, que pode atingir um bom desempenho em termos de métricas relacionadas a cada camada, mas não são otimizados para maximizar o desempenho global da rede e minimizar o consumo de energia e a taxa de transferência dos sensores (AKYILDIZ, 2006; ALMEIDA et al., 2010). Diante disso, (MELODIA et al., 2005) destaca como uma alternativa promissora a criação de protocolos que utilizem o relacionamento cruzado entre camadas, denominado projeto *cross-layer*.

Outra alternativa bastante empregada na literatura é o desenvolvimento de protocolos de roteamento baseados no modelo hierárquico, em que a rede é organizada em *clusters.* Esses protocolos diferenciam-se entre si pela maneira como ocorre à formação de *clusters.* O uso de arquiteturas baseadas em *clusters* apresenta uma redução no consumo de energia e maximiza o desempenho global da rede, além de permitir a inserção de uma quantidade maior de nós e uma maior cobertura da rede (BRUST et al, 2008).

1.2 Objetivos

Diante das restrições das RSSFs e das possíveis soluções encontradas na literatura, este trabalho tem como objetivo apresentar o resultado do desenvolvimento de um algoritmo que utiliza o relacionamento cruzado entre camadas e que considera a formação de *clusters* na rede. Com isso, pretende-se minimizar o consumo de energia e garantir uma comunicação com uma quantidade mínima de atraso, além de possibilitar uma maior escalabilidade e melhor cobertura da rede.

Para atingir esse objetivo foi utilizado o algoritmo proposto em Almeida (2010), que relaciona as camadas de aplicação, rede e enlace, em que as camadas de aplicação e rede se relacionam por meio do compartilhamento de informação, enquanto as camadas de rede e enlace são fundidas em uma só camada.

Para esta dissertação é utilizado o protocolo de acesso múltiplo por divisão de tempo (TDMA – *Time Division Multiple Access*) para controlar o acesso ao meio, garantir sincronização e evitar colisões, além de delimitar a área do *cluster* e assegurar que cada *cluster* só poderá possuir uma quantidade máxima de *n* nós. Isso evita a sobrecarga de líderes e a existência de líderes próximos uns dos outros, eliminando possíveis sobreposições.

Assim, ao algoritmo apresentado em Almeida (2010) foi acrescentada a capacidade de formação de *clusters,* a fim de assegurar escalabilidade e economia de energia a RSSF.

1.3 Estrutura da dissertação

Esta dissertação está organizada em mais quatro capítulos, descritos de maneira resumida a seguir:

No Capítulo 2 é apresentado o conceito de redes de sensores sem fio, enfatizando a arquitetura dos nós sensores bem como as características relevantes a este trabalho;

No Capítulo 3 é discutida a importância da interação cruzada entre

camadas e da formação de *clusters* nas RSSFs, em que trabalhos relacionados a estes temas são mencionados;

O Capítulo 4 aborda a proposta de um algoritmo para formação de *clusters* utilizando um protocolo de roteamento com relacionamento cruzado entre as camadas de aplicação, rede e enlace, descrevendo o funcionamento do algoritmo. Neste capítulo também são apresentados os resultados de simulação de acordo com os parâmetros analisados;

E, por fim, no Capítulo 5 são apresentadas as conclusões da dissertação e os possíveis trabalhos futuros decorrentes da pesquisa.

Capítulo 2

Redes de Sensores Sem Fio

2.1 Definição

A crescente evolução dos sistemas micro-eletromecânicos (MEMS – *Micro Electro-Mecanical Systems*), sistemas embarcados e tecnologias de comunicação sem fio proporcionaram o surgimento e a definição do conceito de RSSF (KARL e WILLIG, 2005; XIONG, 2009). Considerada uma nova classe de redes, as RSSFs são compostas de dispositivos com capacidades de sensoriamento que cooperam uns com os outros para realizar tarefas, uma vez que um único nó não seria capaz de abranger uma grande área (KARL e WILLIG, 2005; ALMEIDA, 2010).

Em essência, os dispositivos que compõem uma RSSF possuem a capacidade de computação, comunicação sem fio e de sensoriamento ou controle, e pode interagir com o ambiente controlando ou monitorando seus parâmetros físicos (KARL e WILLIG, 2005; BASAGNI, 2007). Ela possui características como auto-organização, colaboração, processamento em rede, e comunicação centrada nos dados.

Avanços ocorridos nas RSSFs têm possibilitado o desenvolvimento de sensores cada vez menores, de baixo consumo, baixo custo e multifuncionais, como por exemplo, sensores geofísicos e geoquímicos (SOUTO, 2007; ALMEIDA, 2010),

aumentando assim o campo de atuação dessas redes.

Tais redes podem ser utilizadas em vários domínios de aplicações e em ambientes distintos uma vez que são flexíveis e possuem características que as diferem de redes convencionais (TAN e MUNRO, 2007). A Figura 2.1 resume algumas possíveis aplicações em que as RSSFs podem ser encontradas.

Figura 2.1: Exemplo de possíveis aplicações para RSSFs, adaptada de (BRITO, 2009).

O uso do potencial dessas redes tem impulsionado a pesquisa em várias áreas que buscam, dentre outras coisas, garantir qualidade de serviço, tolerância a falhas, escalabilidade, aumento do tempo de vida útil, ampla faixa de densidade, programabilidade e manutenibilidade nas RSSFs.

As seções seguintes explanarão melhor as características e funcionalidades das RSSFs relevantes a este trabalho. Este capítulo está organizado da seguinte maneira: A Seção 2.2 apresenta a arquitetura dos sensores, mostra os componentes e estados de operação dos sensores, e a arquitetura da RSSF com sua pilha de protocolos; algumas características das RSSFs são elencadas na Seção 2.3; e na Seção 2.4 são dadas as considerações finais deste capítulo.

2.2 Arquitetura dos Sensores

Historicamente, muitos dos trabalhos relacionados à arquitetura das RSSFs têm sido realizados no âmbito da auto-organização de redes móveis e *ad hoc*. Embora essas redes se destinem a diferentes fins, elas compartilham a necessidade de uma forma de organização distribuída (KARL et al., 2005).

Uma RSSF é descrita como uma coleção de nós cooperando na coleta de informações em um ambiente e as enviando para um ponto central, chamado nó sorvedouro (*sink)*. Este possui a tarefa de centralizar e processar as informações vindas dos demais sensores da rede. (ALMEIDA et al., 2010). A Figura 2.2 apresenta um exemplo da estrutura de organização de uma RSSF.

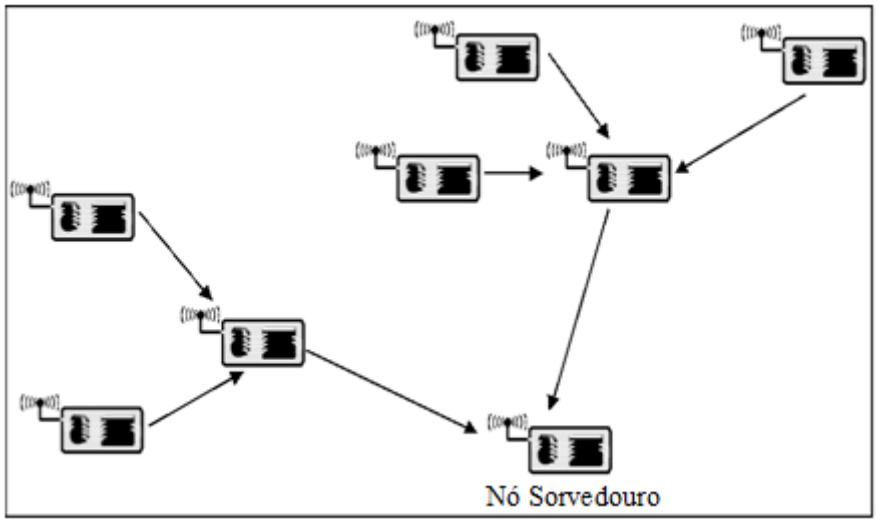

Figura 2.2: Exemplo de organização de uma RSSF.

As RSSFs possuem vários desafios que têm impulsionado diversas propostas de soluções. Segundo Texeira (2005), para uma solução ser eficiente, especialmente em termos de consumo de energia, é necessário a integração de vários aspectos da rede. Desde conhecer o funcionamento do *hardware* que irá compor o nó sensor, a entender as características da arquitetura de *software* da rede. Diante disto, as subseções seguintes abordam as arquiteturas de *hardware* e *software* do nó sensor.

2.2.1 Arquitetura do Nó Sensor

Não existe um padrão comum para definir a arquitetura de um nó sensor, normalmente um modelo geral para um nó sensor é composto de cinco componentes básicos, mostrados na Figura 2.3 (LOUREIRO et al 2003; DELICATO, 2005; KARL e WILLIG, 2005), em que:

a) Controlador – processa todos os dados relevantes, capaz de interpretar instruções de baixo nível e de controlar os demais componentes do sensor;
b) Memória – armazena programas e dados intermediários, geralmente diferentes tipos de memória são usadas por programas e dados;

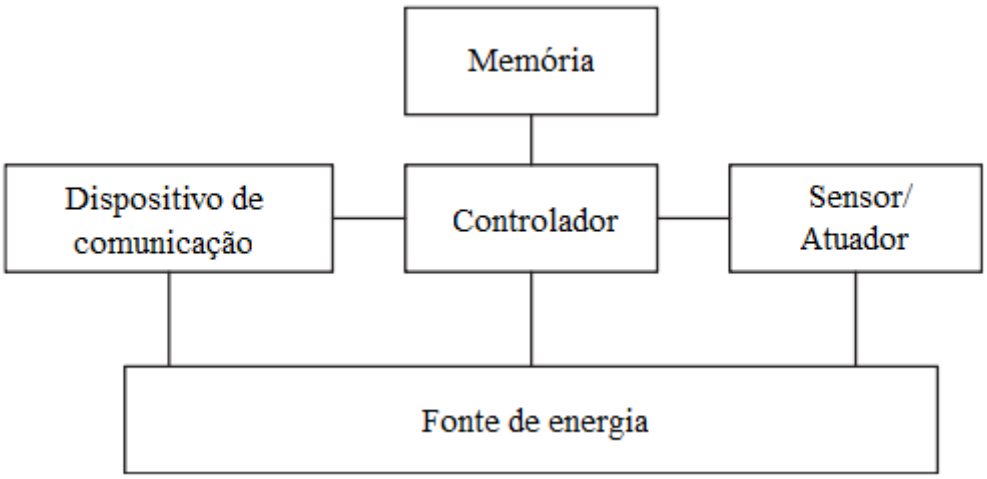

Figura 2.3: Arquitetura de *hardware* do sensor, adaptada de (KARL e WILLIG, 2005).

c) Sensor e/ou atuador – componente responsável pela coleta de dados na RSSF, capaz de observar e controlar parâmetros físicos do ambiente. Existem diferentes tipos de dispositivos de sensoriamento, tais como: acústicos, infravermelhos, sísmicos, térmicos, magnéticos, visuais e radares. Capazes de monitorar uma grande variedade de condições, incluindo: temperatura, movimento, pressão, umidade, níveis de ruídos, entre outros;
d) Dispositivo de comunicação – responsável por receber e enviar informação sobre o canal sem fio, realizando a comunicação entre os nós. Normalmente utiliza comunicação via radiofrequência por meio de um *transceiver* ou radiotransmissor que em geral pode operar em três modos: recepção,

transmissão e desligado;

e) Fonte de energia – componente que fornece energia ao nó sensor para seu funcionamento, normalmente essa fonte é uma bateria não recarregável e/ou de difícil substituição.

Dependendo da aplicação um nó sensor pode possuir outros componentes como, por exemplo, GPS *(Global Positioning System),* fontes de alimentação alternativas, acelerômetros, entre outros (ALKYILDIZ et al, 2002; BORGES NETO, 2009). A Figura 2.4 apresenta dois modelos de *hardware* de sensores.

Figura 2.4: Modelos de sensores de luz, temperatura e som, adaptada de (HURRAY, 2011).

Em geral, o tempo de vida de um nó sensor está diretamente ligado ao tempo de duração de sua bateria. Com isso, é necessário que os outros componentes de *hardware* do sensor usem de forma adequada essa fonte de energia. É importante mencionar que os grandes consumidores de energia dos sensores são os dispositivos de comunicação e de controle (ALMEIDA, 2010).

Como forma de utilizar melhor essa fonte de energia e prolongar o tempo de vida útil da RSSF, um nó sensor pode alternar seu estado de ligado para um estado de menor consumo de energia uma vez que os nós não precisam estar ligados o tempo todo (KARL e WILLIG, 2005). Diante disso, Almeida (2010) apresenta três estados de um sensor:

a) Ativo – neste estado o nó possui todos os seus componentes ligados. É nele que o sensor executa tarefas de rede ou de aplicação, como por exemplo, a transmissão ou recepção de dados. Esse estado é responsável pelo maior consumo de energia em um sensor e deve ser alternado de tempos em tempos para estados de menor consumo;
b) Dormência – representa o estado em que apenas os elementos básicos do sensor estão ligados e no qual a energia é conservada, portanto, o sensor deve permanecer nele a maior parte do tempo. Nesse estado o sensor não pode executar tarefas de aplicação nem ser utilizado pela rede;
c) Inativo – neste estado o nó está sem energia e não pode executar nenhuma tarefa, devendo ser substituído ou descartado.

2.2.2 Arquitetura de *Software*

Entre os componentes de *software*, destaca-se a pilha de protocolos usada pelas RSSFs que é composta de cinco camadas: aplicação, transporte, rede, enlace e física, representada pela Figura 2.5.

Figura 2.5: Representação da pilha de protocolos.

Camada de aplicação

A camada de aplicação serve de interface entre o usuário e o sistema, realizando consultas dos sensores ou operações nos dados para que o usuário

possa obter alguma informação relevante (SOUTO, 2007).

A diversidade de aplicações e os requisitos de *hardware* e *software* de cada aplicação tornam o projeto de aplicações para RSSFs uma tarefa desafiadora. Diferentes tipos de *software* devem ser desenvolvidos e usados para interagir com a RSSF, uma vez que os interesses da aplicação indicam o tipo de dado a ser monitorado, a frequência com que deve ser coletado, os requisitos de qualidade de serviço, o tempo em que os dados devem ser transmitidos, dentre outros (TEXEIRA, 2005).

Para este trabalho não é necessário conhecer todos os tipos de aplicações de uma RSSF. São considerados apenas os principais estados da aplicação descritos em Almeida (2010):

a) Leitura de dados do sensor – a aplicação realiza atividades para coleta de informações do ambiente;
b) Agregação de dados – agrega dados dos sensores vizinhos como forma de economizar energia e otimizar a carga da rede.

É importante ressaltar que esses estados da aplicação só podem ocorrer quando o nó sensor encontra-se no estado ativo.

Camada de transporte

A camada de transporte tem a função de manter o fluxo de dados entre a origem e o destino na comunicação em uma mesma rede, garantindo confiabilidade na entrega de dados, controle de congestionamento e vazão justa entre os nós sensores (TEXEIRA, 2005; SOUTO, 2007).

Camada de rede

A camada de rede é responsável pelo roteamento de dados, projetada seguindo os princípios de eficiência em energia, roteamento centrado em dados, agregação de dados, e atributos baseados em rótulos e localização. É desejável que

ela seja tolerante a falhas. Permite a comunicação com outras redes de sensores, sistemas de controle e comando, e Internet (PAZZI, 2004).

Nas RSSFs os protocolos de rede devem estabelecer rotas que aumentem o tempo de vida da rede em detrimento das outras métricas de desempenho. Para estabelecer rotas, eles podem utilizar a escolha de nós com maior energia disponível, nós com menor consumo de transmissão, caminhos mais curtos ou uma combinação destes. Porém, nas RSSFs o roteamento deve considerar as restrições estabelecidas pelas demais camadas da pilha de protocolos para garantir rotas eficientes em consumo de energia (SOUTO, 2007). Essa camada pode, por exemplo, usar serviços da camada de enlace para entrega de pacotes e como auxílio em operações de roteamento e controle de topologia. Esta é fundamentalmente a idéia do uso do relacionamento cruzado entre camadas.

A camada de rede possui três tarefas que merecem destaque, são elas:

a) Transmissão – quando o nó está transmitindo alguma informação;
b) Recepção – quando o nó está recebendo alguma informação;
c) Dormência – quando não há tarefas a serem executadas.

Nota-se que assim como as tarefas de aplicação, as tarefas de rede devem ser executadas no período ativo do nó. Com isso, torna-se importante realizar essas tarefas de maneira sincronizada evitando assim um consumo desnecessário de energia. Outra questão importante a ser considerada é que tarefas de transmissão e recepção são as maiores consumidoras de energia e, portanto, deve-se minimizar a quantidade de dados trafegados pela rede.

Camada de enlace

A camada de enlace é responsável pela confiabilidade do enlace obtida por mecanismos de controle de erros, pelo protocolo de acesso ao meio utilizado para evitar ou diminuir o número de colisões entre nós vizinhos e pelo enquadramento dos dados, determinando o tamanho de quadro a ser transmitido para assegurar uma transmissão confiável com um mínimo de sobrecarga.

Em suma, a principal tarefa dessa camada é criar um enlace de comunicação confiável entre os nós vizinhos (KARL e WILLIG, 2005). Com isso, muitos dos trabalhos relacionados à camada de enlace são voltados para o desenvolvimento de protocolos de controle de acesso ao meio (MAC – *Medium Access Control*) que auxiliem os nós na decisão de quando e como acessar o canal. Segundo Souto (2007) e Almeida (2010), o compartilhamento de informações dessa camada pode ser útil ao processo de decisão das demais camadas.

Camada Física

A camada física é responsável pela transmissão de *bits*, pela seleção da frequência e pela modulação e codificação de dados digitais, esta última tarefa realizada pelos tranceptores. Tais tarefas são dependentes da eficiência energética do projeto da camada física. Um grande desafio nas RSSFs é encontrar esquemas de modulação e arquiteturas de tranceptores que sejam simples, de baixo custo e, ao mesmo tempo, robustas o suficiente para fornecer o serviço desejado.

Alguns dos pontos cruciais que influenciam a concepção da camada física em RSSF são: baixo consumo de energia, que tem como consequência a pequena potência de transmissão e, portanto, um alcance de transmissão de pequeno porte. A maioria dos componentes de *hardware* deve ser desligada ou operar em modo de baixo consumo a maior parte do tempo. Comparativamente, é necessária baixa taxa de dados, baixa complexidade de implementação e custos, e baixo grau de mobilidade.

2.3 Características das RSSFs

As RSSFs possuem características próprias que as diferem das redes de computadores tradicionais. Muitas dessas características são provenientes das necessidades de suas aplicações ou das tecnologias que as constituem. A seguir são apresentadas características consideradas relevantes a este trabalho.

2.3.1 Limitações

Dentre as principais características das RSSFs deve-se mencionar o limitado poder de processamento, de armazenamento e em especial a limitada capacidade de energia. As restrições computacionais existentes nos nós sensores são os principais fatores que impulsionam a necessidade de diversas adaptações no *software* destes dispositivos.

A redução no tamanho dos sensores trouxe consigo uma limitação na capacidade de energia dos nós, o que acarretou também uma redução da capacidade de processamento, do espaço de armazenamento e da potência de transmissão, uma vez que estes compartilham a mesma fonte de alimentação (VITORINO, 2006).

A restrição de energia, também pode estar relacionada ao fato de que em muitas aplicações os sensores são colocados em áreas inóspitas ou de difícil acesso, impossibilitando assim a manutenção desses elementos. Na maioria das aplicações, o tempo de vida útil de um sensor depende da quantidade de energia disponível, que geralmente equivale à vida útil de sua bateria.

Diante disso, um projeto de solução para RSSF deve considerar o consumo de energia, visto que energia é a principal limitação existente nos sensores (LOUREIRO, 2003).

2.3.2 Escalabilidade

A quantidade de sensores presentes em uma rede pode variar de acordo com o contexto em que a aplicação está inserida. Uma RSSF pode possuir um grande número de nós e as arquiteturas empregadas e protocolos devem ser capazes de suportar este número. Os esquemas a serem utilizados precisam ser flexíveis o suficiente para suportar grandes ou pequenas quantidades de nós sensores (KARL e WILLIG, 2005).

Esta flexibilidade está vinculada diretamente à tolerância a falhas (SILVA, 2006). O grande número de sensores defeituosos em uma aplicação pode causar mudanças nas características de escalabilidade de uma região da RSSF, que

deve ser re-configurada automaticamente pela rede.

A escalabilidade também pode ser usada para garantir a redundância de informação na RSSF. Uma vez que algumas aplicações exigem mecanismos de segurança, em que o tráfego de informações redundantes é necessário para que a precisão da informação coletada seja melhorada, para isso aumenta-se a quantidade de nós na rede.

2.4 Considerações Finais

Em geral as RSSFs operam em ambientes remotos onde não há a intervenção humana direta, o que exige mecanismos de auto-gerenciamento, possuem restrições de energia, já que sua fonte de energia normalmente se dá por meio de baterias não recarregáveis ou de difícil substituição. Essas características tornam a eficiência em energia nas RSSFs um importante tópico de pesquisa e o tempo de vida da RSSF um parâmetro primordial para o desempenho dessas redes.

Uma questão importante a ser considerada no projeto de uma solução eficiente em energia para RSSFs é o fato da comunicação consumir mais energia que o processamento. Assim, o número de transmissões e recepções devem ser o menor possível. Outro ponto importante é que os nós devem permanecer no estado ativo a menor quantidade de tempo, já que este é o estado que mais consome energia. Como neste estado são realizadas as tarefas de rede e aplicação, a sincronização entre essas duas camadas pode levar a uma melhora significativa no quesito eficiência energética da RSSF.

Capítulo 3

Interação Cruzada entre Camadas e Formação de *Cluster*

3.1 Trabalhos Relacionados

Como discutido nos capítulos anteriores, as RSSFs possuem limitações que têm motivado a criação de soluções que buscam melhorar o desempenho e o tempo de vida dessas redes. Dentre as soluções, destaca-se o uso da interação cruzada entre camadas e da formação de *clusters*, com isso este capítulo aborda esses dois tópicos apresentando características e trabalhos relacionados.

3.1.1 Interação Cruzada entre Camadas

Existem na literatura algumas questões consideradas importantes para as RSSFs, das quais se destacam a necessidade de melhorar a eficiência em energia e de prolongar o tempo de vida da rede (LIANG et al., 2007). Com o intuito de atender a essas questões, muitos trabalhos têm sido desenvolvidos, em comum à maioria deles está o uso da tradicional arquitetura em camadas baseada no modelo OSI (*Open Systems Interconnection*), em que cada camada é vista como uma

entidade separada. Na qual, as soluções são projetadas considerando apenas uma camada individual. Essas soluções proporcionam melhoras no desempenho da rede, no entanto, alguns estudos têm revelado que esse ganho ainda pode ser otimizado utilizando a interação cruzada entre camadas (SOUTO, 2007; ALMEIDA, 2010).

A interação cruzada entre camadas, também conhecida como projeto *cross-layer,* possibilita a comunicação direta entre camadas adjacentes ou não, permitindo que uma camada possa se relacionar com as camadas que estão acima ou abaixo dela. Alguns trabalhos, por exemplo, abordam o relacionamento entre as camadas de rede, MAC e física; MAC e física; transporte e física; transporte e rede; MAC, rede e aplicação; entre outros [CUI, 2007; LIANG, 2007; SOUTO et al., 2007; ALMEIDA, 2011]. A Figura 3.1 ilustra possíveis relacionamentos cruzados entre camadas.

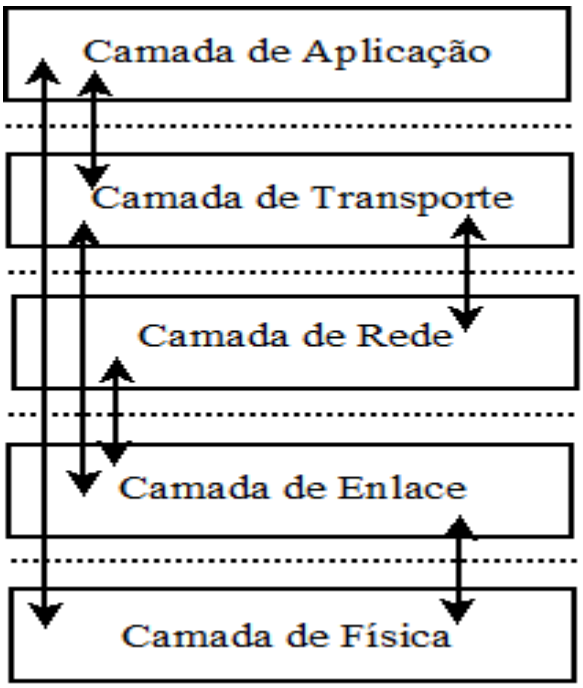

Figura 3.1: Exemplo de possíveis relacionamentos cruzados entre camadas, adaptada de (SOUTO, 2007).

Ao utilizar uma abordagem com relacionamento cruzado entre camadas é importante pensar na melhor forma de incluir acesso à informação, interpretar e representar dados, e maneiras de como alterar a informação (SOUTO, 2007). Muitos dos projetos nessa área dividem a interação em duas categorias: o

projeto integrado entre duas ou mais camadas, conhecido como união de funcionalidades; e a proposta de uma nova arquitetura utilizando o compartilhamento de informações (ALMEIDA, 2010). A seguir são apresentados alguns trabalhos que utilizam a interação cruzada entre camadas e que são relevantes para esta dissertação.

Em Souto et al. (2007) é proposto um projeto integrado entre as camadas de rede, acesso ao meio e aplicação, cujo objetivo é mostrar que a comunicação direta entre essas camadas pode melhorar o desempenho e o tempo de vida das RSSFs. Nesse projeto as rotas são estabelecidas de acordo com informações de energia dos nós na camada física e de conectividade dos nós na camada de enlace. Com o conhecimento das rotas, a camada de aplicação só envia mensagens quando está certa do caminho fim a fim. Nessa proposta, a camada de aplicação consulta a camada de rede para saber se há uma rota para o destino desejado, caso não haja rota disponível, o dado é armazenado no *buffer* da aplicação até que uma rota seja estabelecida ou que a validade do dado expire.

A abordagem proposta em Souto et al. (2007) usa as informações da camada B-MAC, uma variação do protocolo MAC, para reduzir a quantidade de mensagens de controle e tornar eficiente o processo de descoberta e manutenção de vizinhança. Tal proposta apresenta uma diminuição no consumo de energia por *byte* transmitido, uma vez que a aplicação conhece a rota, melhorando assim a taxa de entrega da rede e consequentemente diminuindo o consumo de energia. De forma geral, essa arquitetura otimiza a taxa de entrega de pacotes, minimiza o atraso fim a fim, diminui a perda de pacotes e, por conseguinte, o processo de retransmissão.

Outra abordagem estudada é a apresentada em Van Hoesel et al. (2004), que propõe uma arquitetura em que o protocolo de acesso ao meio se baseia no TDMA, que possui fatias de tempo de tamanho fixo, e se relaciona com o protocolo de roteamento. Nessa abordagem, o sensor é capaz de escolher sua fatia de tempo baseado apenas na informação local, operando independente da estação base. Essa escolha é estudada em conjunto com o algoritmo de roteamento e o ciclo de trabalho, para que os nós possam se comunicar uns com os outros livres de colisão e para que nós pertencentes ao caminho traçado pelas rotas tenham o ciclo

de trabalho sincronizado com o algoritmo. Os resultados de simulações mostram que essa abordagem consegue um ganho significativo no tempo de vida da rede quando comparado a protocolos tradicionais, como o SMAC (*Sensor MAC*) e o DSR (*Dynamic Source Routing*) (Van Hoesel et al., 2004). Um exemplo de cenário de rede para este algoritmo é apresentado na Figura 3.2.

No estudo realizado por Cui (2007) é proposta uma arquitetura, cujo objetivo é a união das camadas de rede, MAC e física, que contribui para minimizar o consumo de energia total da rede, explorando a relação entre transmissão e consumo de energia dos circuitos. Utiliza especificamente o TDMA como esquema de controle de acesso ao meio e otimiza o fluxo de roteamento atribuindo fatias de tempo TDMA, taxa de modulação e energia na camada de enlace.

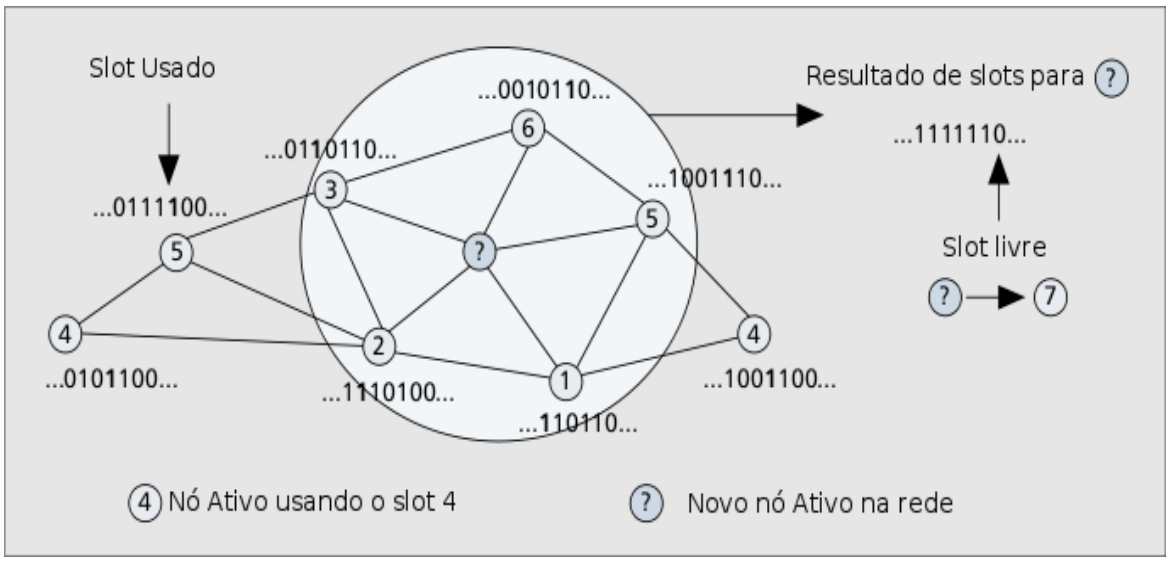

Figura 3.2: Cenário de rede, adaptada de (VAN HOESEL, 2004).

Por meio de exemplos numéricos Cui (2007) mostra que é possível uma significativa economia de energia quando se utiliza a adaptação de enlace. Esse projeto, assim como os citados anteriormente, apresentou uma diminuição no consumo de recursos da rede se comparado a protocolos baseados na tradicional arquitetura em camadas, provando que abordagens que utilizam a interação cruzada entre camadas podem trazer ganhos relevantes às RSSFs.

3.1.2 Formação de Cluster

A formação de *clusters* em RSSFs é uma tarefa realizada pelo protocolo de roteamento, que determina quais nós serão líderes e com qual líder cada nó se comunicará para transmitir seus dados. Segundo Lazzarotto (2008), o roteamento de dados nas RSSFs apresenta vários desafios, dos quais se pode destacar:

a) A dificuldade das RSSFs utilizarem protocolos baseados em IP;
b) A necessidade das RSSFs encaminharem dados coletados de múltiplas fontes a um determinado ponto de coleta, chamado nó sorvedouro;
c) O fato das RSSFs serem orientadas a aplicação, ou seja, o desempenho da rede está fortemente ligado ao problema a ser resolvido. O que dificulta a criação de um modelo padrão, uma vez que formas diferentes de tratar o roteamento devem ser consideradas;
d) A redundância de dados, normalmente ocasionada pela grande quantidade de sensores instalados em um ambiente. Cabe ao protocolo de rotemanto tratar essa redundância e evitar a transmissão desnecessária de dados, minimizando o gasto de energia.

Com isso, vários trabalhos foram propostos para tratar o roteamento em RSSF. Em geral, o roteamento pode ser classificado em plano ou hierárquico. No primeiro, a comunicação é realizada por meio de múltiplos saltos, cada nó executando a mesma tarefa. Enquanto, no segundo os nós são agrupados em grupos, chamados *clusters,* e a comunicação com o nó sorvedouro é realizada apenas por nós específicos (LAZZAROTO, 2008). As Figuras 3.3 e 3.4 apresentam respectivamente, a estrutura do protocolo de roteamento plano e do protocolo de roteamento hierárquico.

Para esta dissertação será considerado apenas o roteamento hierárquico, uma vez que o uso de arquiteturas organizadas em *clusters* reduz o consumo de energia e maximiza o desempenho da rede. Permite a realização eficiente de protocolos de roteamento e MAC, agregação de dados e mecanismos

de segurança (BRUST et al, 2008).

Nas RSSFs, um *cluster* representa um grupo de nós interconectados, com um nó dedicado chamado líder (*cluster head*) responsável pelo gerenciamento de *cluster*, escalonamento de acesso ao meio, disseminação de mensagens de controle e agregação de dados, que pode levar a um balanceamento de carga na rede (LIU et al., 2009).

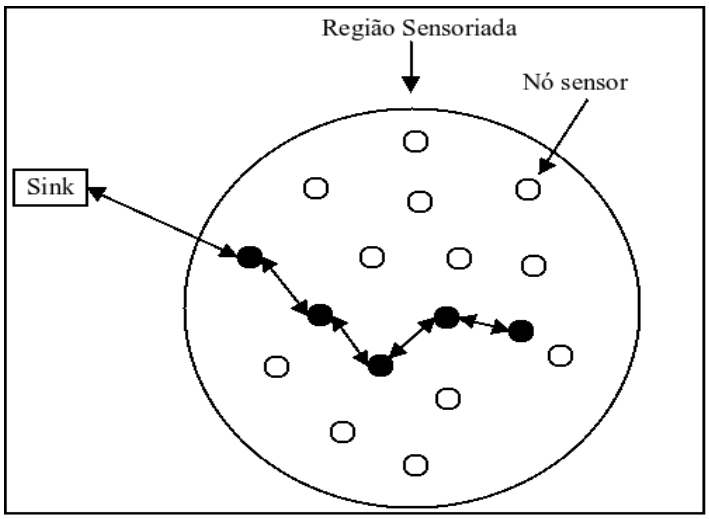

Figura 3.3: Roteamento plano, adaptada de (LAZZAROTO, 2008).

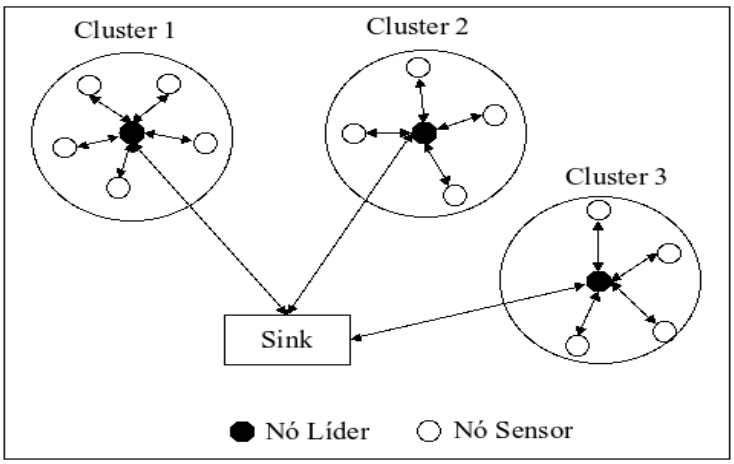

Figura 3.4: Roteamento hierárquico.

Em geral, a maioria dos protocolos de roteamento hierárquico realiza uma eleição para decidir quais nós na rede serão líderes de *clusters.* De acordo com o método utilizado para seleção, o algoritmo de formação de *cluster* pode ser

classificado em centralizado ou distribuído (LIU et al., 2009).

O algoritmo centralizado pode gerar *clusters* justos, que apresentam quantidades de nós e consumo de energia semelhantes, porém, é adequado apenas para redes pequenas, uma vez que neste o processamento em *cluster* precisa conhecer a informação global da rede. O algoritmo distribuído, por sua vez, é adequado para redes de grande escala que não precisa deter a informação global da rede, porém é mais difícil controlar a estrutura do *cluster* (LIU et al., 2009). Na literatura há uma grande quantidade de trabalhos que propõem algoritmos de roteamento em RSSF organizadas em *clusters,* para esta dissertação foram estudados alguns que são apresentados a seguir.

LEACH

O LEACH (*Low-Energy Adaptative Clustering Hierarchy*) é um protocolo auto-organizável e de agrupamento adaptativo que altera de tempos em tempos seus líderes, capaz de realizar computação local em cada *cluster* e reduzir a quantidade de dados que devem ser transmitidos ao nó sorvedouro. Proporcionando uma redução na dissipação de energia, já que nas RSSFs o processamento gasta menos energia que a comunicação (HEINZELMAN, 2000).

No LEACH, a criação de *clusters* é feita conforme os seguintes passos:

1. Define *a priori* a quantidade de nós líderes que a rede deve possuir;
2. Escolhe aleatoriamente os nós líderes de cada rodada, considerando o conjunto de nós que ainda não foram líderes e o número de vezes que cada nó já foi líder;
3. Cada líder envia uma mensagem de anúncio aos demais nós na rede;
4. Cada nó não líder decide a qual *cluster* irá pertencer, se tornando membro do *cluster* cujo sinal de anúncio, enviado pelo líder, tenha sido recebido com maior intensidade. E então informa ao líder do *cluster* que é membro de seu grupo;
5. Baseado no número de nós por *cluster* e no número de nós que gostaria de pertencer a seu *cluster*, o líder cria um escalonamento TDMA que é difundido

para os nós no *cluster,* informando a cada nó quando ele deve transmitir dados.

Uma vez criados os *clusters* e feito o escalonamento TDMA, o LEACH inicia a fase de transmissão de dados. Esse protocolo assume que os nós sempre têm dados a enviar, porém esse envio só ocorre durante o tempo de transmissão alocado pelo líder para cada nó membro de seu *cluster.* Essa transmissão utiliza uma quantidade mínima de energia e o rádio dos nós não líderes pode ser desativado quando não estiver no tempo alocado para transmissão, economizando energia da rede (HEINZELMAN, 2000).

Em suma, o LEACH apresenta uma quantidade de nós e área conhecidas, nó sorvedouro dedicado, agregação de dados e uma seleção de líderes aleatória em que é calculado um número ótimo de líderes. Em cada rodada, nós que ainda não foram líderes participam da eleição de líderes, isso acontece até que todos os nós tenham em algum momento sido líder e então o processo de eleição reinicia.

ACE

O ACE (*Algorithm for Cluster Establishment*) é um algoritmo de roteamento que busca selecionar o menor número de líderes de forma que todos os nós na rede pertençam a um *cluster.* Assim, fornecem uma cobertura eficiente da rede e minimizam a sobreposição de *clusters.* Isto reduz a quantidade de contenção de canal entre os *clusters*, além de melhorar a eficiência de algoritmos como roteamento e agregação de dados, que são executados no nível dos líderes (CHAN e PERRING, 2004).

O ACE consiste de duas partes lógicas, a criação de novos *clusters* e a migração de *clusters* existentes. Novos *clusters* são gerados em um processo de auto-eleição, quando um nó decide tornar-se líder, ele envia uma mensagem de difusão para recrutar seus vizinhos, que se tornarão membros do novo *cluster*. Um nó pode ser membro de mais de um *cluster* enquanto o protocolo estiver executando, e apenas no final do protocolo ele se torna membro de um único *cluster*.

A migração de um *cluster* existente é controlada pelo líder. Cada líder

periodicamente faz uma seleção entre todos os seus membros (por exemplo, seus vizinhos) para determinar qual é o melhor candidato para tornar-se o novo líder do *cluster.* O melhor candidato é o nó que caso seja líder passe a ter o maior número de seguidores, minimizando a sobreposição com *clusters* existentes. Uma vez escolhido o melhor candidato, o líder atual irá promover o melhor candidato como novo líder e abdicar da sua posição de líder. Assim, a posição do *cluster* irá migrar em direção ao novo líder fazendo que alguns dos seguidores do líder antigo deixem de pertencer ao *cluster* e que novos nós se tornem seguidores do novo líder.

O ACE opera independente do tamanho da rede e sem precisar conhecer a posição geográfica do nó ou qualquer tipo de distância ou estimativa de direção entre os nós, esses fatores são considerados vantagens desse protocolo.

HETCP

Continuando os trabalhos sobre criação de *clusters*, Kejun et al. (2006) apresentaram o HETCP (*A Hierarchical Energy Efficient Topology Control Protocol*) um protocolo que combina métricas de balanceamento e conservação de energia de forma a otimizar a topologia da RSSF, além de considerar as restrições de energia, operação autônoma e escalabillidade. A idéia principal desse protocolo é estender o esquema *active-sleep* à RSSF hierárquica. O HETCP considera viável admitir que os nós sensores são implantados de modo redundante e os organiza em *clusters* locais de modo a distribuir a carga de energia. Dentro dos *clusters,* um subconjunto de nós é selecionado como ativos, enquanto outros são mantidos dormindo para economia de energia. A Figura 3.5 apresenta uma arquitetura de rede para o protocolo HETCP.

No HETCP existem duas fases, a fase de formação de *clusters* e a de escalonamento de estado dos nós. Para minimizar a sobrecarga essa primeira fase é curta e não se repete até que um *cluster* falhe (KEJUN et al., 2006).

Na fase de formação de *clusters* ocorre a escolha dos nós que serão líderes baseada na energia atual e no número de vizinhos de cada nó. Após essa escolha, os líderes transmitem uma mensagem de anúncio aos demais nós. Cada nó não líder junta-se ao *cluster,* cujo sinal transmitido pelos líderes seja recebido mais forte, e então informa ao líder que tornou-se membro de seu grupo. Após essa

fase, a RSSF será composta de *clusters,* cada um com um líder responsável por agregar dados dos nós em seu grupo e transmiti-los diretamente ao nó sorvedouro. Nessa fase, pode ocorrer a reformulação de *clusters* caso a energia residual do líder seja menor que o limiar de energia permitido ou caso esgote sua energia.

Por sua vez, na fase de escalonamento ocorre a mudança de estado dos sensores, que pode ser:

a) Ativo – os nós realizam o sensoriamento do ambiente e enviam os pacotes de dados para o líder do *cluster*. Uma mensagem de ajuda é anunciada se a energia residual de um nó ativo estiver abaixo do limiar permitido*;*

b) Escuta – os nós sondam a rede para decidir se eles necessitam trabalhar ou dormir;

c) Dormindo – os nós desligam a maioria de seus componentes, deixando apenas o relógio ligado para acordar mais tarde.

Quando comparado ao LEACH, o HETCP apresentou um tempo de vida duas vezes maior e a fração de energia residual do sistema é sempre maior que no LEACH (KEJUN et al., 2006).

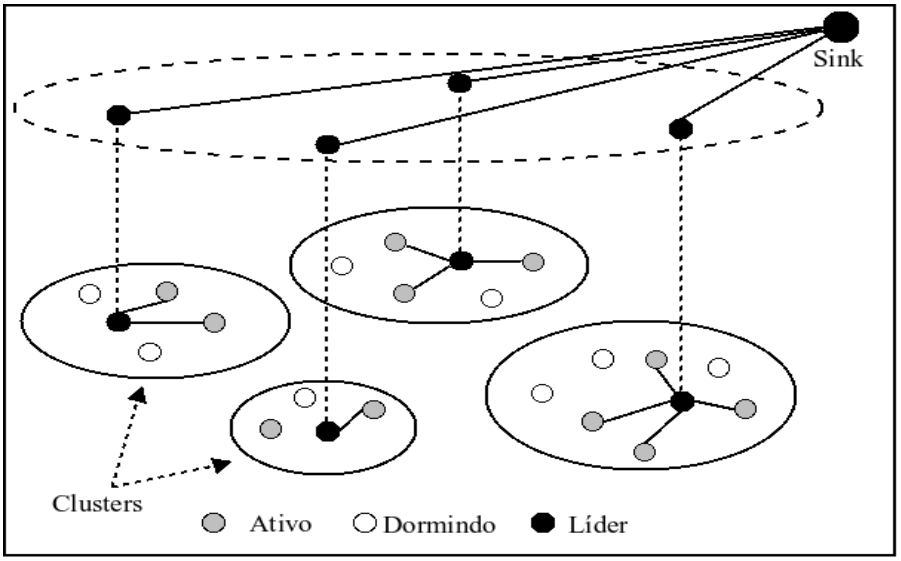

Figura 3.5: Arquitetura de rede do HETCP, adaptada de (HEJUN et al., 2006)

KCCTC

A pesquisa realizada por Meng et al. (2008) discutiu um mecanismo de controle de topologia baseado na criação de *clusters* em RSSF, o KCCTC (*K-Connected Cluster Topology Control Protocol*). O principal objetivo desse trabalho foi a formação de *clusters*, bem como a escolha dos nós líderes e dos nós *gateways*, do escalonamento de estados e otimização de *cluster.* Nesse algoritmo os nós sensores podem ser classificados em líder, *gateway*, *gateway* distribuído e nó membro. A Figura 3.6 apresenta a topologia de RSSF do KCCTC.

O KCCTC inclui três principais partes que são a formação de *clusters*, a otimização e ajuste de *cluster,* e o escalonamento de estados dos nós. Na primeira fase é realizada a seleção dos nós que serão líderes, *gateways* e *gateways* distribuídos. Na segunda, é realizado o ajuste do *cluster* considerando dois aspectos principais o tratamento de *clusters* isolados e o aumento da redundância geral do *cluster.* Na terceira, são escalonados os estados dos nós sensores que podem ser: descoberta, nó *backbone* dormindo, nó comum dormindo e ativo.

Percebe-se que a seleção de nós líderes e o escalonamento de estados dos nós são cruciais às redes de sensores agrupadas em *clusters,* pois permitem uma melhor cobertura da rede e reduzem o consumo de energia nos nós.

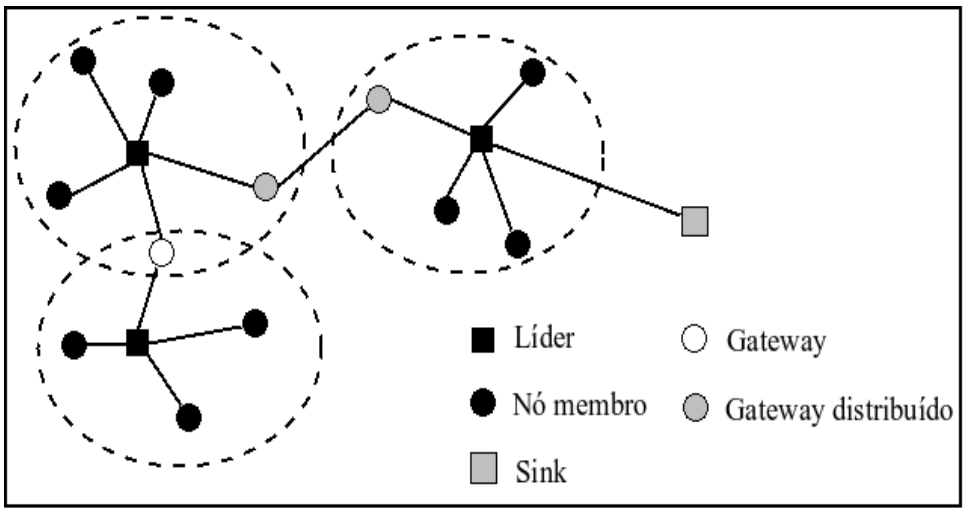

Figura 3.6: Topologia de RSSF do KCCTC, adaptada de (MENG et al.; 2008)

Além dos trabalhos mencionados anteriormente, também foi estudado

o trabalho realizado por Liang (2007), que propõe uma arquitetura de otimização que utiliza o relacionamento cruzado entre as camadas MAC e física em uma RSSF organizada em *cluster.* No artigo não é informado como é realizada a eleição dos líderes, mas sabe-se que os líderes possuem uma capacidade de energia e processamento superiores aos demais nós da rede e que os nós sensores usam o esquema TDMA para transmitir suas informações ao líder, eliminando possíveis interferências na comunicação.

Na abordagem de Liang (2007) a camada física detém a informação do estado do canal que será usada pelos nós líderes no algoritmo de fusão de dados. De acordo com as diferentes informações do canal dos nós sensores e dos requisitos de taxa de erro de *bit* do nó líder, considera-se o algoritmo de escalonamento de transmissão na camada MAC. O algoritmo usa o número de *bits* que os nós sensores podem transmitir para substituir seu tempo de vida correspondente e desenvolver uma matriz de escalonamento para escalonar o período de transmissão dos sensores.

Esse protocolo, assim como outros cujas redes são organizadas em *clusters*, o líder atua como gerente do *cluster* com responsabilidade de controlar e gerenciar os sensores envolvidos em seu *cluster*. Os resultados de simulações mostram que o uso do esquema de transmissão proposto em Liang (2007) ajuda a minimizar o gasto de energia nos nós sensores e, consequentemente, a prolongar o tempo de vida da rede.

3.2 Considerações Finais

Neste capítulo foram apresentadas duas características importantes para esta dissertação, a interação cruzada entre camadas e a criação de *clusters* em RSSF. O uso da primeira consegue um ganho significativo em termos de energia prolongando o tempo de vida da rede, enquanto o uso da segunda, além de influenciar positivamente no consumo de energia, garante escalabilidade e uma maior cobertura da rede.

Capítulo 4

Algoritmo para Formação de Conjunto de Sensores em RSSF utilizando Relacionamento Cruzado entre Camadas

Para este trabalho será utilizada a proposta apresentada em Almeida (2010), cujo objetivo é o desenvolvimento de um algoritmo que relaciona as camadas de aplicação, rede e enlace, em que se busca otimizar o período de atividade dos nós sensores, uma vez que soluções que procuram otimizar apenas uma única camada apresentam uma deficiência ocasionada pela separação entre camadas, o que pode ser observado ao comparar os ciclos de trabalho da camada de aplicação e das camadas de rede e enlace, na Figura 4.1.

Diante disto, Almeida (2010) propôs que as tarefas realizadas pela camada de aplicação e pelas camadas de rede e enlace tenham seus ciclos de trabalhos ajustados e sincronizados, como ilustrado na Figura 4.2.

Nesse algoritmo, o relacionamento entre as camadas de aplicação e de enlace se dá por compartilhamento de informações, utilizando o protocolo de

controle de acesso ao meio TDMA como forma de garantir a sincronização e evitar colisões entre vizinhos. Já as camadas de enlace e de rede se relacionam por meio do desenvolvimento conjunto por completa união de funcionalidades.

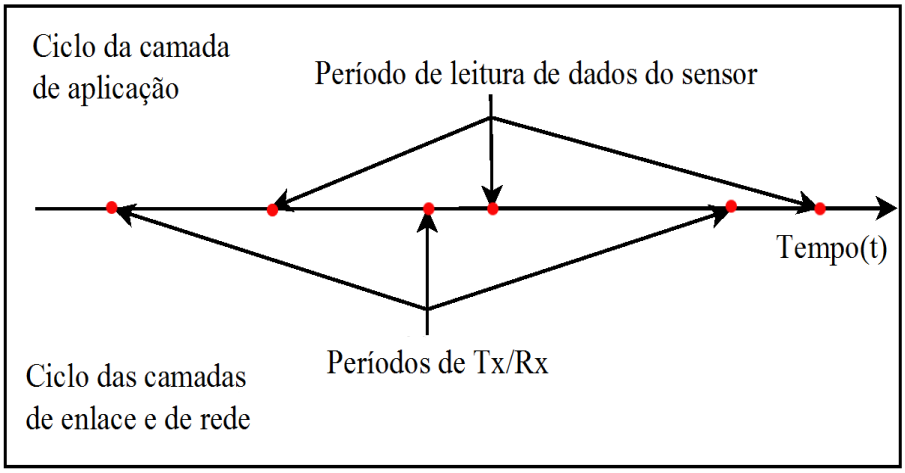

Figura 4.1: Comparação dos ciclos de trabalho das camadas de aplicação, rede e enlace.

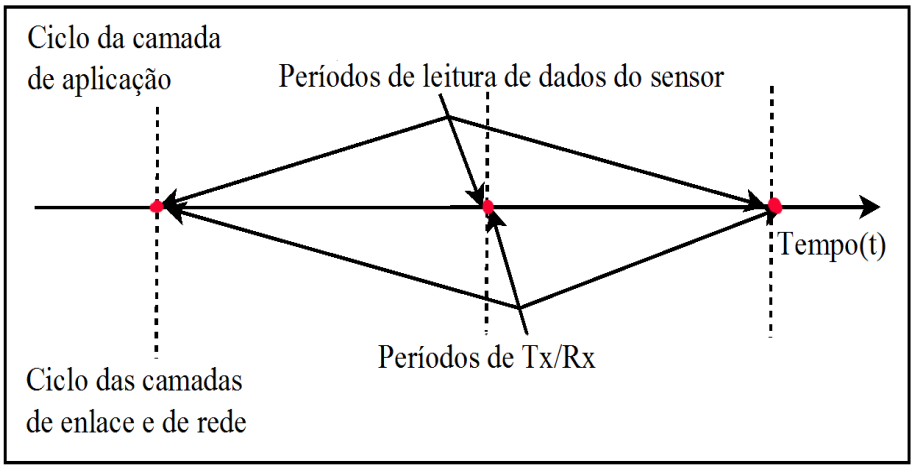

Figura 4.2: Tarefas de aplicação sincronizadas com as de rede e enlace.

O sincronismo surgiu neste trabalho como forma de melhorar o desempenho com relação ao atraso que na camada de aplicação pode ser agravado pela falta de sincronia com as camadas mais baixas. Com isso, o esquema TDMA é usado pela camada de enlace para oferecer referência de tempo em que os eventos de transmissão e recepção poderão acontecer. Assim, o relacionamento da camada de enlace com a camada de aplicação permite a sincronia da aplicação com as

camadas mais baixas, de modo que antes de executar as tarefas de transmissão e recepção a aplicação realiza suas atividades e somente então são executadas as tarefas de rede.

Desse modo, a aplicação flexibiliza a execução de suas tarefas para se adaptar ao tempo mais conveniente de acordo com as informações dos intervalos da camada de enlace, no qual as outras camadas também estarão prontas a executar.

O modelo TDMA utilizado nessa abordagem possui intervalos de tempo de tamanhos variáveis para que nós próximos ao nó sorvedouro tenham intervalos de tempo maiores do que nós mais afastados, usados como ponderação pelo algoritmo de roteamento para formar rotas. O relacionamento entre camadas proposto em Almeida (2010) é ilustrado na Figura 4.3.

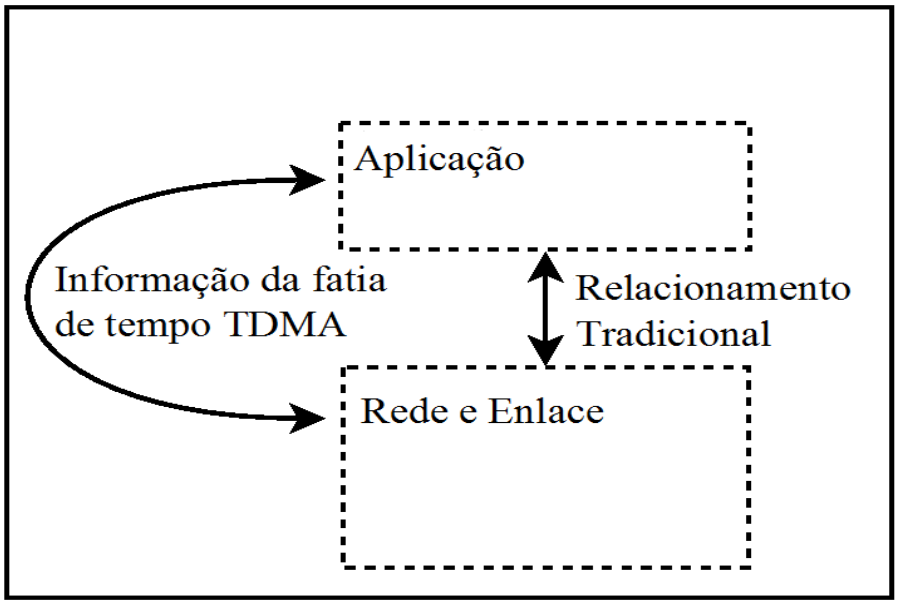

Figura 4.3: Compartilhamento de informação entre a aplicação e as camadas de rede e enlace.

Os resultados de simulações numéricas sugerem que o uso do algoritmo proposto por Almeida (2010) minimiza o atraso médio na RSSF, por executar as tarefas de aplicação e rede de maneira sincronizada, escolhendo o momento mais adequado para gerar tráfego na rede. Além disso, apresenta um ganho relevante em energia, devido à sincronia existente entre a aplicação e a camada de rede que evita mudanças de estado desnecessárias para as tarefas da

aplicação. Além disso, há o escalonamento da camada de rede com a camada enlace, para que ao terminar um período de recepção de dados de outro nó, este possa receber os dados de um terceiro nó vizinho.

Apesar de tais vantagens, esse algoritmo possui uma limitação quando o assunto é escalabilidade, pois a quantidade de nós na rede é limitada à quantidade de ciclos que o protocolo TDMA tem a oferecer. Assim, quando não há mais fatias de tempo a distribuir entre os nós vizinhos não é possível inserir outros nós à rede, limitando o número de nós que podem fazer parte da rede.

Esta dissertação teve o objetivo de dotar o algoritmo proposto por Almeida (2010) da capacidade de formação de conjunto de sensores ou *clusters* à RSSF, de maneira a garantir escalabilidade e cobertura de rede, e reduzir o atraso.

4.1 Descrição do Algoritmo

Como visto no Capítulo 3, nas RSSFs um *cluster* representa um grupo de nós interconectados com um nó dedicado chamado líder (*cluster head)* responsável pelo gerenciamento do *cluster*, escalonamento de acesso ao meio, disseminação de mensagens de controle e agregação de dados (LIU et. al., 2009).

Com o objetivo de garantir escalabilidade, economia de energia e reduzir o atraso fim a fim à RSSF, este trabalho propõe um algoritmo de formação de *clusters* em que a rede é dividida em *clusters* ou conjuntos de nós sensores. Para isso, a abordagem de Almeida (2010) é utilizada como algoritmo de formação de *cluster,* assumindo que o nó sorvedouro neste caso passe a agir como líder de um *cluster*.

Para este trabalho é usada como referência a arquitetura de nó apresentada na Figura 4.4. A arquitetura do nó sensor é alterada para conseguir o relacionamento cruzado entre as camadas de aplicação, rede e enlace.

Assim como os demais algoritmos de formação de *clusters,* este também possui duas fases: a formação de *clusters* e o escalonamento de estados. Na fase de formação é realizada a escolha dos líderes e é delimitada a área de cada

conjunto de sensores. Para isto, é utilizada a seguinte inequação

$$n(p/b) < d < c/m, \quad (1)$$

em que, *n* é o número máximo de nós em cada segmento de rede; *b* é a taxa de transferência entre dois nós; *p* é o tamanho máximo do pacote de dados; *c* é o tamanho do ciclo TDMA; *m* é o número máximo de vizinhos para os quais um nó pode aceitar rotear dados; e *d* é quantidade de fatias de tempo que o nó utilizará para transmitir dados.

A utilização dessa abordagem permite a delimitação da área do *cluster*, assegurando que cada *cluster* só poderá possuir uma quantidade *n* de nós, evitando a sobrecarga de líderes e a existência de líderes próximos uns dos outros e eliminando possíveis sobreposições.

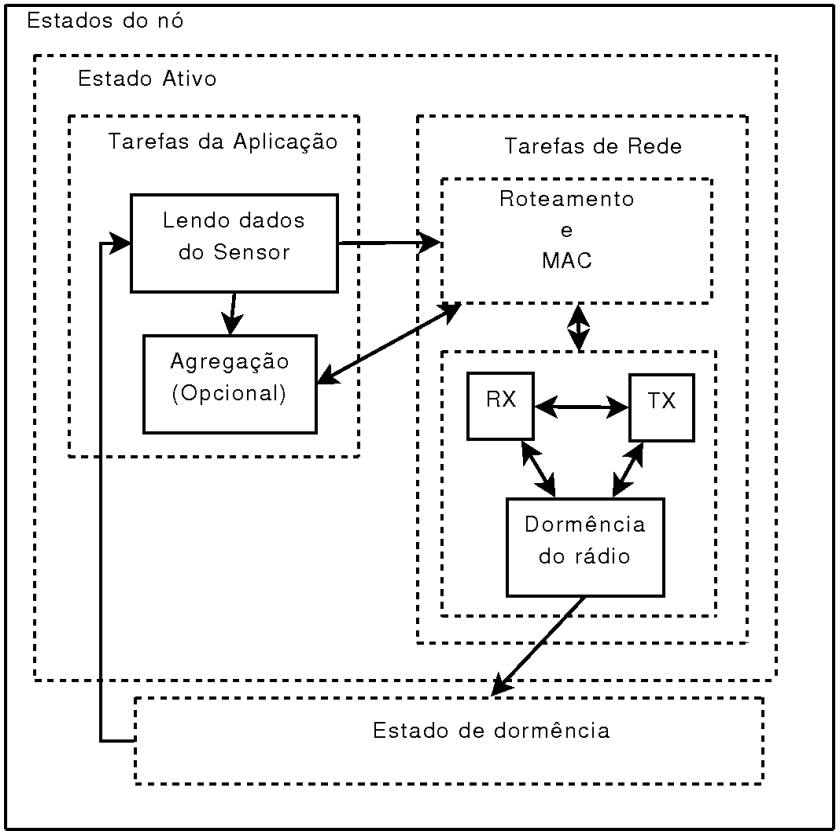

Figura 4.4: Arquitetura do nó.

Neste algoritmo, a formação de *cluster* é baseada na distribuição de fatias de tempo TDMA. Os passos são descritos a seguir e representados na Figura 4.5. Na Figura 4.5 têm-se no eixo X os períodos de tempo em que acontece o envio e o recebimento de mensagens e no eixo Y a disposição dos nós sensores que enviam e recebem mensagens. Por exemplo, no tempo *T0* tem-se o nó *Líder* enviando uma mensagem chamada *Anuncio* para os nós que estão ao seu alcance, neste caso os nós *1* e *2*, já no tempo *T1* tem-se o envio de uma mensagem do tipo *Tentativa* pelo nó *1* para o nó *Líder,* e assim por diante.

A formação de *clusters* se dar de acordo com os seguintes passos:

1. Descoberta de vizinhos e Controle de Topologia
    a. Um nó é escolhido *a priori* como líder. Ele deve então iniciar a comunicação enviando uma mensagem em modo difusão *(broadcast),* chamada “Anuncio”. Nessa mensagem o nó informa o tamanho da fatia de tempo que tem a oferecer e a fatia de tempo livre;
    b. Os nós 1 e 2 recebem a mensagem do líder e usam o algoritmo de controle de topologia para decidir se irão se comunicar, ou não, diretamente com o líder. Nesse momento somente o nó 1 irá se conectar ao líder.

2. Competição por uma fatia de tempo
    a. O nó 1 envia uma mensagem chamada “Tentativa” para o nó líder com sua identificação;
    b. O nó líder usa o algoritmo de alocação de fatia de tempo para escolher a fatia a ser alocada, e então envia uma mensagem chamada “Oferta” com a fatia de tempo para o nó 1 se comunicar com o líder.

3. Repasses
    a. O nó 1 recebe a mensagem de “Oferta” e reescalona as suas atividades de acordo com a fatia de tempo fornecida pelo líder;
    b. Agora o nó 1 tem a função de dar conectividade a outros nós. Para isso

ele deve calcular o seu valor de *d;*

c. O nó 1 reinicia o processo de alocação de fatia de tempo a partir do passo 1-a, no entanto, o tamanho da fatia de tempo oferecida está subtraído do valor *d* calculado pelo nó 1.

4. Formação de *clusters*

a. Os passos do Item 1-a até o Item 3-c se repetem até que não haja mais fatias de tempo disponíveis para alocação. Nesse momento o nó 4 envia uma mensagem em modo difusão, chamada “SemSlot”. Nessa mensagem o nó 4 informa a seus vizinhos que não possui fatia de tempo a oferecer e, caso eles ainda não façam parte da rede, um novo *cluster* deve ser criado;

b. Os nós 2, 3, 5 e 6 recebem a mensagem “SemSlot” do nó 4. O nó 2 e 3, que já pertencem a um *cluster* descartam a mensagem “SemSlot”;

c. Nesse momento o nó 5 torna-se líder de *cluster*, e então muda seu tipo de nó comum para nó líder e recebe um intervalo de tempo a oferecer de tamanho igual ao que o nó líder *a priori* recebeu;

d. O nó 5 inicia o processo de alocação de fatias de tempo seguindo os passos a partir do Item 1-a, até que todos os nós da rede façam parte de um *cluster*.

Na fase de escalonamento de estados ocorre a mudança de estado dos nós sensores, que pode ser alterado para ativo, dormindo ou inativo. Esses estados são semelhantes aos presentes no algoritmo HETCP, apresentado no Capítulo 3.

O escalonamento TDMA realizado pelos líderes para formar e delimitar *clusters*, também é utilizado para informar a cada nó quando ele deve transmitir dados. Essa última característica está presente também no protocolo LEACH, diferindo apenas no fato que no LEACH todos os nós recebem fatias de tempo de tamanhos semelhantes, e não de tamanho variável como apresentado neste trabalho.

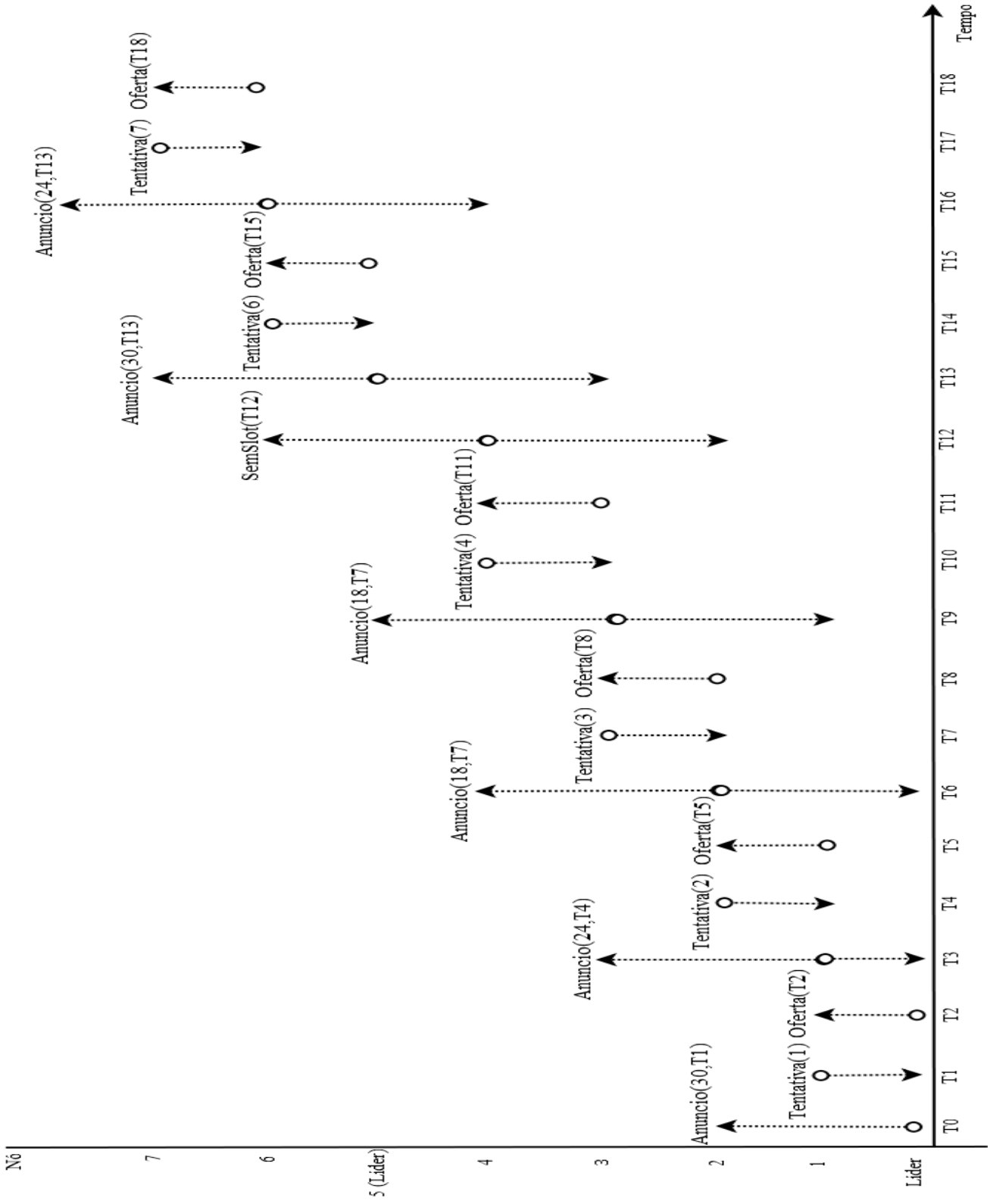

Figura 4.5: Alocação de intervalos de tempo

4.2 Estados do Protocolo

Nesta seção são descritos os estados do protocolo representados pela Figura 4.6, a Figura 4.5 é utilizada para ilustrar os eventos:

1. Todos os nós iniciam, mas como nenhum deles possui uma rota, ficarão no estado “Aguarda evento”;

2. O nó líder *a priori*, ao iniciar, calcula o valor de *d* e logo depois envia a mensagem “Anuncio”, T0 na Figura 4.5;

3. O nó líder então passa para o estado “Aguardar Tentativas”. O nó 1 recebe um evento e segue para o estado “Recebi Anúncio” e posteriormente para “Tentar Associação”, T1 na Figura 4.5;

4. O nó líder recebe a “Tentativa” do nó 1, e continua processando até o estado “Enviar Alocações”, T2 na Figura 4.5;

5. O nó 1 recebe o evento e segue para “Recebi Alocação” e “Aceitar Rota”, e reinicia o ciclo, T3 na Figura 4.5;

6. Esse processo se repete até que não haja mais fatias de tempo disponíveis para alocação. Nesse momento o nó envia uma mensagem de difusão “SemSlot”, T12 na Figura 4.5;

7. O nó 5 recebe a mensagem “SemSlot”, e então torna-se líder, calcula o valor de *d* e logo depois envia a mensagem “Anuncio”, T13 na Figura 4.5.

8. A partir desse momento um novo *cluster* surge na rede e os estados de 1 a 7 passam a se repetir em tempos diferentes, até que todos os nós façam parte da rede.

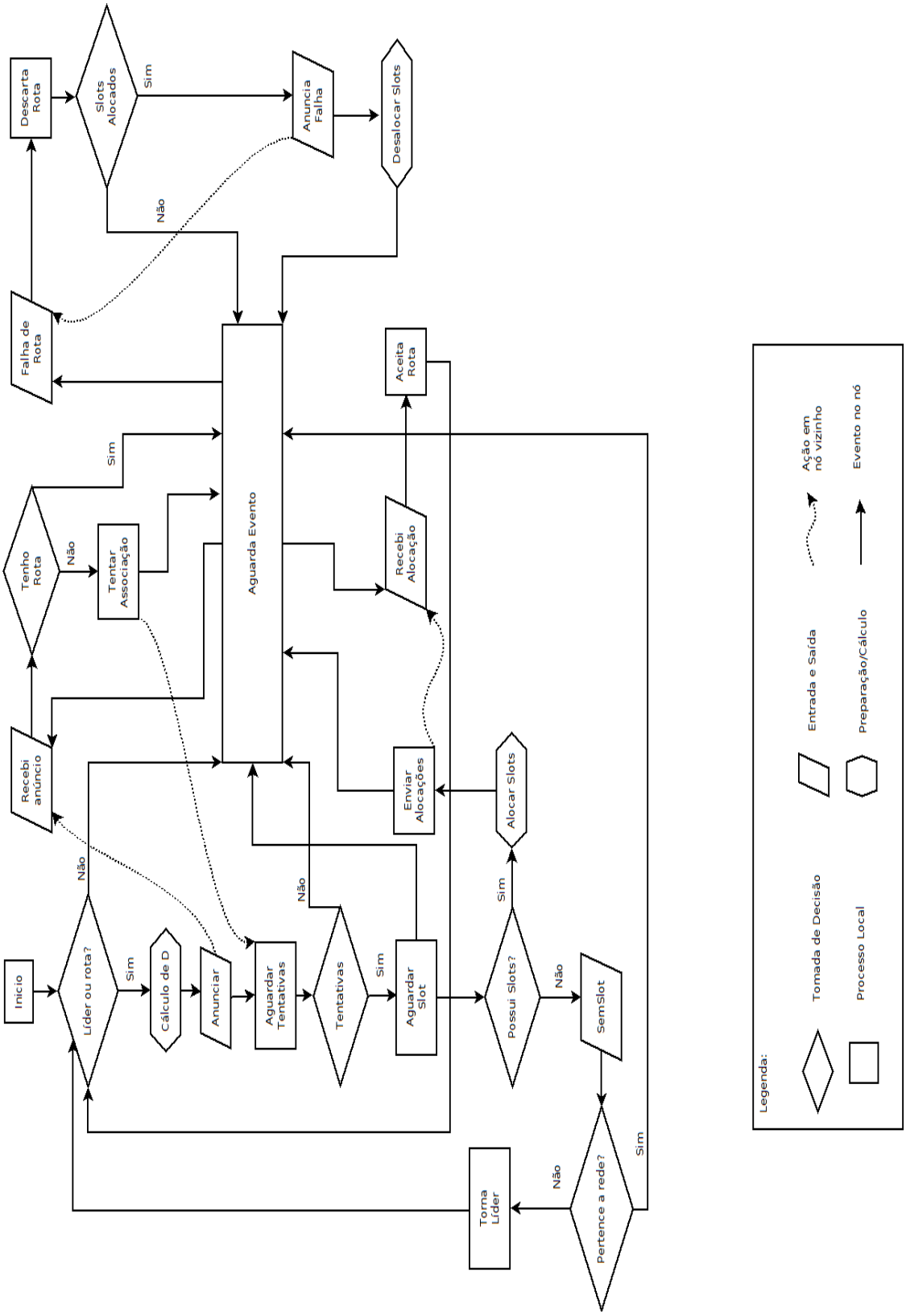

Figura 4.6: Estados do Protocolo

4.3 Resultados

4.3.1 Ambiente de Simulação

Para avaliar o modelo proposto foi utilizado o simulador de eventos discretos OMNeT++ (OMNET MANUAL, 2011), em conjunto com o *Mobility Framework*. Nas simulações todos os nós sensores são fontes de eventos gerados em períodos regulares de tempo. Ao ser gerado um evento a camada de aplicação encaminha os dados a serem transmitidos pelas camadas de rede. Todos os nós são homogêneos e todos os relógios (*clocks)* estão sincronizados por fatias de tempo TDMA. A topologia usada na simulação é mostrada na Figura 4.7.

Para haver um termo de comparação para o modelo proposto, é simulada a abordagem proposta em Almeida (2010). Porém, para simular uma rede com a mesma quantidade de nós do modelo é necessário aumentar o tamanho do ciclo TDMA.

Nas simulações são usados os seguintes parâmetros:

a) Tempo total de simulação: 360 segundos;
b) Periodicidade: 4 segundos;
c) Simula-se um radio transmissor de b= 25 k*bit*/s;
d) Tamanho de pacotes da camada de aplicação de *p*=64 *bits*;
e) Número máximo de nós em um segmento de rede *n*=15;
f) Número máximo de vizinhos que um nó poderá rotear *m*=5;
g) Tamanho de um ciclo TDMA *c*=2.1 segundos na rede com formação de *clusters* e *c*=4.2 segundos na rede sem formação de *clusters.*

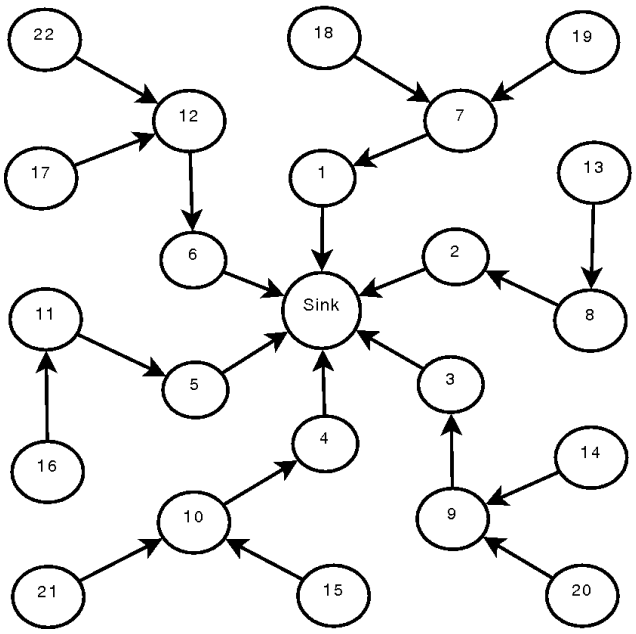

Figura 4.7: Topologia de rede.

4.3.2 Métricas Analisadas

Para avaliar o desempenho do algoritmo proposto são analisadas métricas como escalabilidade, atraso da rede e energia.

Escalabilidade

Como visto no Capítulo 2, a escalabilidade está diretamente ligada à quantidade de sensores presentes na rede, que pode variar de acordo com o contexto em que a aplicação está inserida. Uma RSSF pode possuir um grande número de nós e as arquiteturas empregadas e protocolos devem ser capazes de suportar este número. Os esquemas utilizados precisam ser flexíveis o suficiente para suportar grandes ou pequenas quantidades de nós sensores (KARL e WILLIG, 2005). Para analisar a escalabilidade do algoritmo proposto, são utilizados os parâmetros anteriores, e mais os seguintes:

a) Quantidade inicial de fatias de tempo 30 na rede com *cluster* e 50 na rede sem *cluster;*

b) E, a quantidade de nós presentes na rede assume os seguintes valores: 7, 10, 13, 15 e 20 nós. Isso para averiguar se os resultados se mantêm estáveis.

Para essa avaliação é analisado o atraso médio na rede, como mostrado na Figura 4.8. Pode-se observar que o atraso apresentado na abordagem com formação de *clusters* é inferior ao apresentado na proposta de Almeida (2010).

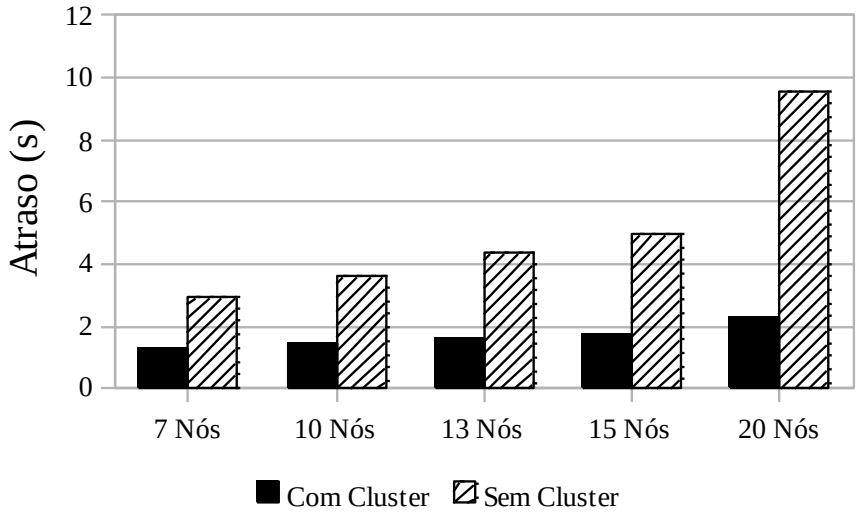

Figura 4.8: Comparação do atraso médio na rede variando a quantidade de nós.

Atraso

A segunda métrica analisada é o atraso ou latência que representa o intervalo de tempo entre a produção da informação e a chegada da informação ao destino final. Para RSSF a importância dada ao atraso depende da aplicação. Por exemplo, em aplicações do tipo agricultura de precisão pode-se tolerar algum atraso, já em aplicações de monitoramento crítico o atraso deve ser evitado. O atraso é visto em (BISDIKIAN et. al., 2010) como um dos critérios para avaliar a qualidade da informação (QoI – *Quality of Information*), ou seja, embora o atraso seja tolerável, a presença dele faz com que a informação sensoriada tenha sua validade expirada.

Para avaliar o atraso apresentado pela abordagem proposta são utilizados os parâmetros apresentados na Subseção 4.3.1, os resultados das simulações são analisados nas Figuras 4.9, 4.10 e 4.11.

Na Figura 4.9 o atraso médio da rede, com e sem formação de *clusters*, é comparado. Na Figura 4.10 é apresentado o atraso médio em cada nó na rede com e sem *clusters*. Em ambos os casos pode-se perceber um ganho no atraso da abordagem com *cluster*s em relação à abordagem prévia de Almeida (2010).

Na Figura 4.11 é apresentado o atraso em cada nó, especificando a qual *cluster* cada nó pertence, em que $N_{i,j}$ representa o *i*-ésimo nó do *j*-ésimo *cluster*. Analisando o gráfico pode-se observar que o atraso nos *clusters* é semelhante, o que não acontece no caso da rede sem *cluster* mostrado na Figura 4.10. Ou seja, o nó $N_{1,1}$ possui uma latência da mesma ordem da apresentada pelos nós $N_{1,2}$ e $N_{1,3}$, e assim por diante, o que caracteriza a equidade (*fairness*) entre os nós da rede.

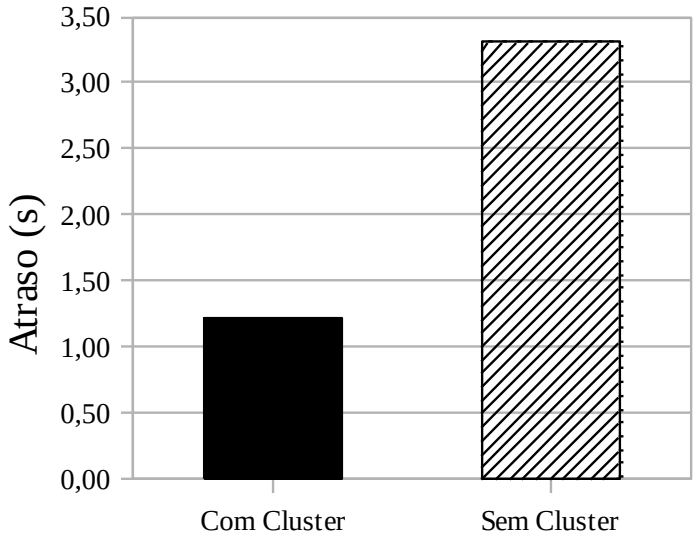

Figura 4.9: Comparação do atraso médio na rede.

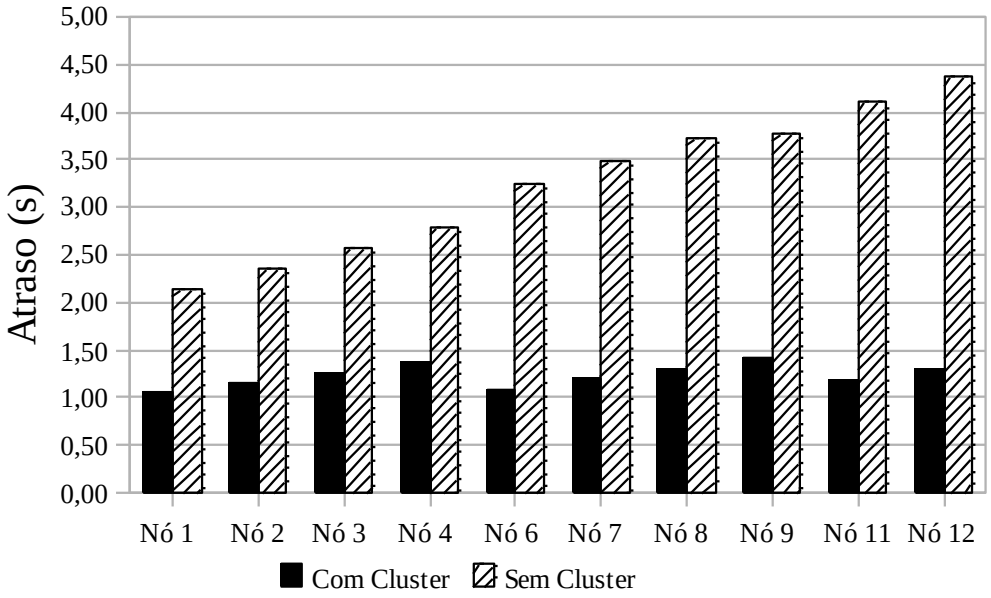

Figura 4.10: Comparação do atraso médio em cada nó.

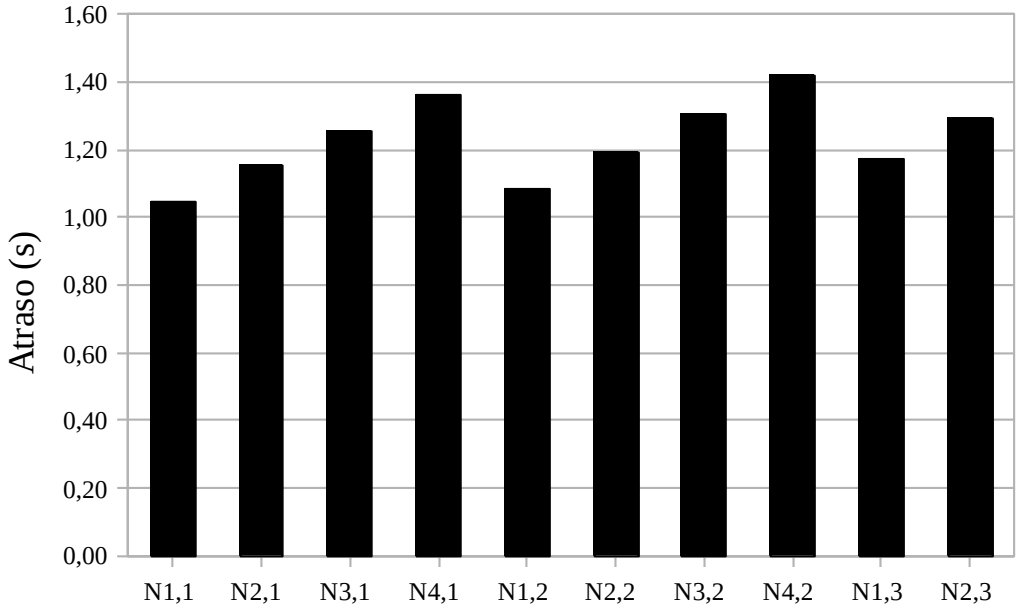

Figura 4.11: Atraso médio para cada nó nos *clusters*.

Variando a Periodicidade

Nesta seção é avaliado o atraso na RSSF com *cluster* variando o fator

periodicidade, que é o intervalo de tempo entre a geração de eventos na camada de aplicação. E pode também ser encarado como o padrão de tráfego na rede, que neste caso seria tráfego uniforme. A avaliação da periodicidade é representa pela Figura 4.12, em que se pode notar que o atraso diminui conforme o período entre duas atividades de sensoriamento aumenta. Sendo semelhante ao que ocorre na rede sem *cluster*, Figura 4.13, no entanto, na rede com *cluster* o atraso é menor.

Para essa avaliação foram utilizados os parâmetros apresentados na Subseção 4.3.1, diferindo apenas no seguinte parâmetro:

a) Periodicidade: 2, 4, 8, 16 e 32 segundos

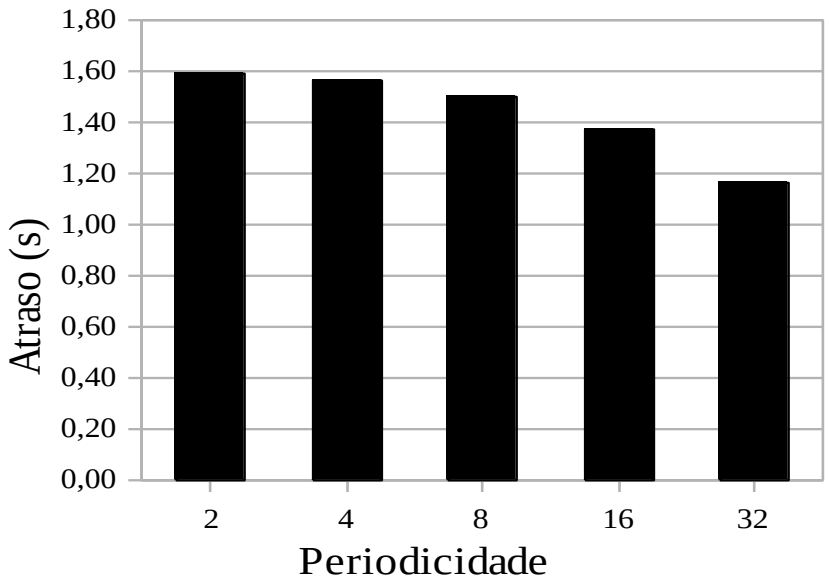

Figura 4.12: Atraso variando a periodicidade na rede com *cluster*

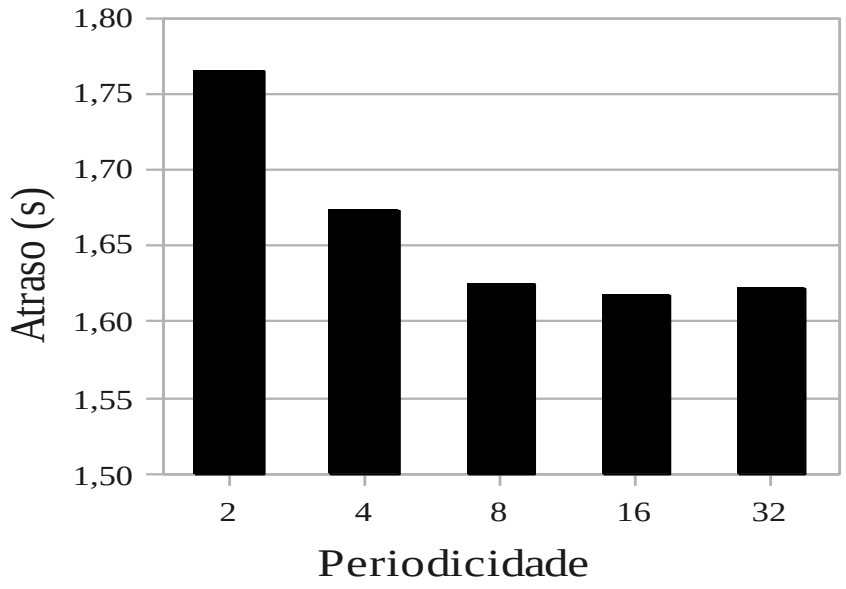

Figura 4.13: Atraso variando a periodicidade na rede sem *cluster* (ALMEIDA, 2010).

Energia

Como mencionado no Capítulo 2, as RSSFs possuem várias restrições das quais destaca-se a energia, visto que os nós sensores são alimentados por uma bateria, muitas vezes não recarregável ou de difícil substituição. Na maioria das aplicações o tempo de vida do nó sensor está diretamente ligado ao tempo de vida de sua bateria, assim um projeto de solução para RSSF deve considerar o consumo de energia como um ponto chave.

Para avaliar o consumo de energia são utilizados os parâmetros da Subseção 4.3.1 em conjunto com os seguintes:

a) Energia consumida pelo radio transmitindo = 8 mW;
b) Energia consumida pelo radio recebendo = 8 mW;
c) Energia consumida pelo radio em estado de dormência = 1 mW;
d) Quantidade inicial de fatias de tempo 30 na rede com *cluster* e 50 na rede sem *cluster;*
e) Tempo de simulação: 6, 12, 18, 24, 30, 36 minutos.

Na Figura 4.14 pode-se observar uma diminuição no gasto de energia quando comparada com a abordagem proposta em Almeida (2010). E na Figura 4.15 é analisado o consumo de energia em cada nó, especificando a qual *cluster* cada nó pertence, em que $N_{i,j}$ representa o *i*-ésimo nó do *j*-ésimo *cluster*. Pode-se observar que o consumo de energia em *clusters* distintos é semelhante, como já esperado.

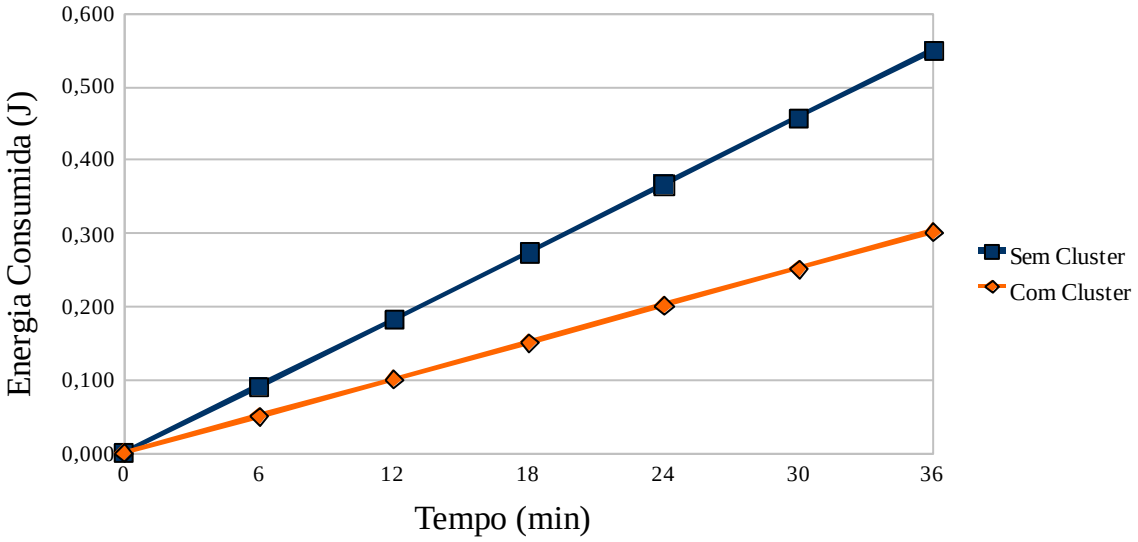

Figura 4.14: Média de energia gasta no 1º nó da rede.

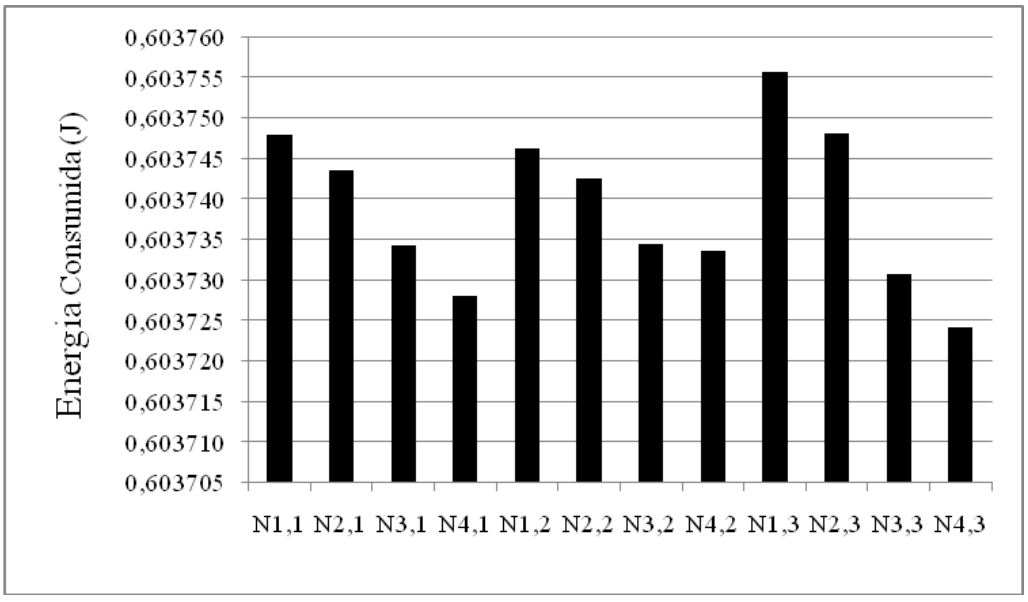

Figura 4.15: Gasto de energia na rede com criação de *cluster*

4.4 Considerações Finais

Neste capítulo, pode-se observar que o uso do relacionamento cruzado entre as camadas de aplicação, rede e enlace como algoritmo para formação e delimitação de *clusters* pode ser considerada uma alternativa às restrições de energia presentes nas RSSFs, visto que os resultados de simulações apontam uma redução no consumo de energia da rede e no atraso fim a fim, quando comparados a abordagem de Almeida (2010), além de garantir escalabilidade.

O diferencial desse trabalho é combinar as vantagens apresentadas pelo relacionamento cruzado entre camadas com as vantagens presentes na organização em *clusters*.

Capítulo 5

Conclusões e Trabalhos Futuros

Conhecidas as limitações existentes nas RSSFs, várias soluções são apresentadas na literatura como forma de contornar essas limitações. Dentre as quais, destaca-se o uso do relacionamento cruzado entre camadas e o uso de protocolos de roteamento seguindo o modelo hierárquico, baseado em *clusters*. Tais soluções têm melhorado o desempenho da rede e o ganho no consumo de energia da rede.

A proposta desta dissertação consistiu no desenvolvimento de um algoritmo de roteamento para RSSF, baseado em *clusters,* que utiliza o relacionamento cruzado entre camadas relacionando as camadas de aplicação, rede e enlace, como apresentado em Almeida (2010). Para este trabalho foi utilizada uma variante do protocolo de acesso ao meio TDMA para formar e delimitar *clusters,* restringindo o número de nós comuns associados a cada líder. Assim, almeja-se balancear o número de nós em cada *cluster*, evitando que um determinando líder possua muitos nós associados a ele, eliminando possíveis sobrecargas e também sobreposições de líderes.

Simulações foram realizadas e parâmetros como escalabilidade, consumo de energia e atraso foram analisados. Ao comparar os resultados obtidos

no algoritmo de formação de *clusters* com relacionamento cruzado entre as camadas de aplicação, rede e enlace, com os resultados alcançados na abordagem proposta em Almeida (2010), observou-se que na primeira há uma diminuição no atraso médio fim a fim na rede e também no consumo de energia, proporcionando um maior tempo de vida aos nós sensores e a rede. Além de garantir escalabilidade ao permitir que uma quantidade maior de nós possa pertencer à RSSF, o que possibilita um aumento da área de cobertura da rede.

Com isso, conclui-se que a abordagem apresentada nesta dissertação, que combina relacionamento cruzado entre camadas e formação de *clusters,* é uma alternativa promissora para o ganho de desempenho em uma RSSF, usando melhor os recursos e aumentando o tempo de operabilidade de tais redes. Porém, esta proposta só pode ser utilizada em “Aplicações Orientadas a Eventos”, devido à exigência de sincronismo com a camada de aplicação.

Como possíveis perspectivas de trabalhos futuros decorrentes desta pesquisa, têm-se:

a) Implementar e analisar outros protocolos de roteamento com formação de *clusters,* conhecidos na literatura, e comparar o desempenho destes com o algoritmo aqui proposto;
b) Estudar atributos da camada física, como por exemplo, codificação, modulação e balanço de potência;
c) Estudar o impacto da mobilidade dos nós na rede com relacionamento cruzado entre camadas com formação de *clusters.*

Referências Bibliográficas

AKYILDIZ, I. F.; SU, W; SANKARASUBRAMANIAM, Y; CAYIRCI, E. Wireless sensor networks: a survey. Computer Networks, v. 38, n. 4, p. 393-422, 15 mar. 2002.

AKYILDIZ, I. F.; VURAN M. C.; AKAN, Ö. B. A cross-layer protocol for wireless sensor networks. In: *Proceedings Conference on Information Sciences and Systems*, p.1102-1107, 22-24 mar. 2006. ISBN: 1-4244-0349-9.

ALMEIDA, E. V.; RAMALHO, W. B.; ALENCAR, M. S.; FONSECA, I.E. Cross-Layer Interaction Between the Application, Routing and MAC Layers for Wireless Sensor Networks. In: *International Telecommunication Symposium*, Manaus, Brasil, v. 1, p. 1-5, 2010.

ALMEIDA, E. V. *Relacionamento Cruzado entre Camadas em Redes de Sensores Sem Fio.* Dissertação (Mestrado) – Mestrado em Ciência da Computação pela parceria entre a Universidade do Estado do Rio Grande do Norte e a Universidade Federal Rural do Semiárido, 2010.

BORGES NETO, J. B. *PROST: Um Protocolo de Roteamento Sensível ao Tempo para Redes de Sensores Sem Fio*. Dissertação (Mestrado) – Universidade Federal do Ceará, 2009.

BRITO, L. F. M. N. de. *Análise Comparativa da Vazão e do Consumo de Energia em Redes de Sensores Sem Fio com Topologia Estrela Utilizando o Padrão IEEE 802.15.4 no Modo Beacon-enabled e Nonbeacon.* Monografia (Graduação), Universidade de Brasília, 2009.

BISDIKIAN, W. W. T. H.; GOECKER, C.; TOWSLEY, G., Target tracking with packet delays and losses – QoI amid latencies and missing data. In: *Proceedings of PerCom Workshops,* Mannheim, Germany, p. 93-98, 2010. ISBN: 978-1-4244-6605-4.

BRUST, M. R.; FREY, H.; ROTHKUGEL, S. Dynamic Multi-hop clustering for mobile hybrid wireless networks. In: *Proceedings of the Second International Conference on Ubiquitous information management and communication*, Suwon, South Korea, p. 130-135, 2008. ISBN: 978-1-59593-993-7.

CHAN, H.; PERRIG, A. ACE: An emergent algorithm for highly uniform cluster formation. In: *Proceedings of the First European Workshop on Sensor Networks*, p. 154-171, 2004.

CUI, S.; MADAN, R.; GOLDSHIMITH, Andrea J.; LALL, S. Cross-layer energy and delay optimization in small-scale sensor networks. In: *IEEE Transactions on Wireless Communications*, p. 3688-3699, 2007. ISSN: 1536-1276.

CURY, R. M. *Uma abordagem difusa para o problema de flow-shop scheduling*. Tese (Doutorado), Universidade Federal de Santa Catarina, 1999.

DELICATO, F. C. *Middleware baseado em serviços para redes de sensores sem fio.* Tese (Doutorado), Universidade Federal do Rio de Janeiro, 2005.

GUIDONI, D. L.; MACHADO, M. do V.; MINI, R. A. F.; LOUREIRO, A. A. F. Difusão de Dados Baseada em Atraso e Energia para Redes de Sensores Sem Fio. XXIV *Simpósio Brasileiro de Redes de Computadores,* Curitiba, Paraná, Brasil, 2006. ISBN: 8576690721.

HEINZELMAN, W. R.; CHANDRAKASAN, A.; BALAKRISHNAN, H. Energy-efficient communication protocol for wireless microsensor networks. In: *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, p. 3005-3014, 2000. ISBN: 0-7695-0493-0.

KARL, H.; WILLIG, A. *Protocols and architectures for wireless sensor networks*. University of Paderborn, Alemanha: John Wiley & Sons, 2005. ISBN 0470095105.

KEJUN, D.; XINGSHE, Z.; XINGGUO, Z.; ZHIGANG, L. HETCP: A hierarchical energy efficient topology control protocol for wireless sensor networks. In: *International Conference on Wireless Communications, Networking and Mobile Computing*, p. 1-4, set. 2006. ISBN: 1-4244-0517-3.

LAZZAROTO, P. *Algoritmos de roteamento hierárquicos em redes de sensores sem fio utilizando algoritmos evolutivos para determinação de cluster-head's*. Dissertação (Mestrado), Universidade Tecnológica Federal do Paraná, 2008.

LIANG, Q. Faul-tolerant energy efficient wireless sensor networks: a cross-layer approach. In: *IEEE Military Communications Conference*, Atlantic City, NJ, USA, v. 3, p. 1862 – 1868, 17-20 out. 2005. ISBN: 0-7803-9393-7.

LIANG, Q.; YUAN, D.; WANG, Y.; CHEN, H. A cross-layer transmission scheduling scheme for wireless sensor networks. In: *Journal Computer Communications,* Newton, MA, USA, v. 30, p. 2987-2994, 14-15 out. 2007.

LIU, Y.; GAO, J.; ZHU, L.; ZHANG, Y. A clustering algorithm based on communication facility in WSN. *In: International Conference on Communication and Mobile Computing*, p.76-86, 6-8 jan. 2009. ISBN: 978-0-7695-3501-2.

LÖBBERS, M.; WILLKOMM, D. *A Mobility Framework for OMNeT++ User Manual*. Disponível em <http://mobility-fw.sourceforge.net/manual/index.html>. Acesso em: maio 2010.

LOUREIRO, A. A. F., NOGUEIRA, J. M. S., RUIZ, L. B., MINI, R. A., NAKAMURA, E. F., and FIGUEIREDO, C. M. S. Wireless sensor networks (in portuguese). In: *Proceedings of the 21st Brazilian Computer Networks*, Natal, RN, Brazil, p. 179-226, 2003. Tutorial.

GUIMARÃES, M. C. R. *A implementação de um módulo HIP na ferramenta de simulação OMNeT++ para avaliação de desempenho de handoff.* Dissertação (Mestrado) – Pontifícia Universidade Católica, 2008.

MELODIA, T.; VURAN M. C,; POMPILI D. *The state of the art in cross-layer design for wireless sensor networks*. In: *Proceedings of EuroNGI Workshops on Wireless and Mobility, Springer Lecture Notes in Computer Science* 3883*,* Como, Italy, p.78-92, jul. 2005. ISSN 978-3-540-34025-6.

MENG, Z.; WANG, S.; WANG, Q. Fault tolerant topology control for clustered wireless sensor networks. In: *4th International Conference on Wireless Communications, Networking and Mobile Computing,* Dalian, China, p. 1-5, 12-14 out. 2008. ISBN: 978-1-4244-2107-7.

OMNeT User Manual. Disponível em: <http://www.omnetpp.org/doc/omnetpp40/manual/usman.html >. Acesso em: abril de 2011.

PAZZI, R. W. N. *Especificação e Implementação de um Protocolo Tolerante a Falhas e de Baixa Latência para Redes de Sensores Sem Fio.* Dissertação (Mestrado) – Universidade Federal de São Carlos, 2004.

SILVA, M. D. *Redes de Sensores sem Fio aplicadas em Ambientes Industriais de Petróleo e Gás*. Monografia (Graduação) – Universidade Federal do Rio Grande do Norte, 2006.

SOUTO, Eduardo James Pereira. *Arquiteturas Cross-layer para Redes de Sensores sem Fio.* Tese (Doutorado) – Universidade Federal de Pernambuco, 2007.

SOUTO, E.; ASCHOFF, R.; SADOK, D.; KELNER, J. Projeto cross-layer entre as camadas MAC, roteamento e aplicação em redes de sensores sem fio. In: *XXV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, Belém, Brasil, v. 01, p. 45-58, 2007.

Technologies & Tools @CISTER/IPP-HURRAY. Disponível em <http://www.hurray.isep.ipp.pt/activities/WSN/Technologies.ashx>. Acesso em: maio de 2011.

TEXEIRA, I. *Roteamento com Balanceamento de Consumo de Energia para Redes de Sensores Sem Fio*. Dissertação (Mestrado) – Universidade Federal do Rio de Janeiro, 2005.

VARGA, A.; HORNIG, R. An overview of the OMNeT++ simulation environment. In: *Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, 2008.

VAN HOESEL, L., NIEBERG, T.; HAVINGA, P. J. M. Prolonging the lifetime of wireless sensor networks by cross-layer interaction. *IEEE Wireless Communications,* p. 78-86, dec. 2004. ISSN: 1536-1284.

VITORINO, B. A. D. *Arquitetura para Construção de Software Adaptativo para Redes de Sensores.* Dissertação (Mestrado) – Universidade Federal de Minas Gerais, 2006.

XIONG, L.; WEN, J.; HUANG, W.; YU, M. The Research of Mobile Fusion-nodes Routes for Data Collection in Multi-tier Mobile Wireless Sensor Network. In: *Proceedings of the 2009 First International Workshop on Education Technology and Computer Science*, v. 2, p. 227-232, 7-8 mar. 2009. ISBN: 978-0-7695-3557-9.

ANEXO A – O Simulador OMNet++ e o Mobility Framework

A.1 Descrição

O OMNeT++ (*Objective Modular Network Testbed in C++*) é um simulador de eventos discretos orientado a objetos utilizado para modelagem de redes de comunicação. Possui arquitetura modular permitindo assim que novos módulos sejam acoplados a ele, sempre que novas funções forem necessárias como, por exemplo, a simulação de redes de sensores (OMNET MANUAL, 2011; VARGA e HORNIG, 2008; GUIMARÃES 2008).

Esse simulador foi desenvolvido por Andras Varga e está disponível para as plataformas Linux, Mac OS/X e Windows. Pode ser utilizado por instituições acadêmicas de ensino e pesquisa e por organizações de pesquisa sem fins lucrativos. O OMNeT++ fornece aos usuários direitos semelhantes ao GNU *General Public License,* o que permite que qualquer usuário possa modificá-lo de acordo com seu propósito e também compartilhar suas modificações com a comunidade. Segundo (OMNET MANUAL, 2011), esse simulador pode ser utilizado em diversos domínios de problemas, tais como:

- Modelagem de redes de comunicação com e sem fio;
- Modelagem de protocolos;
- Modelagem de multiprocessadores e outros sistemas distribuídos de *hardware*;
- Validação de arquiteturas de *hardware*;
- Avaliação dos aspectos de desempenho de sistemas de *software* complexos;
- Modelagem e simulação de qualquer sistema onde a abordagem de evento discreto seja apropriada.

A.1.1 Estrutura do OMNeT++

Quanto a estrutura o OMNeT++ consiste de módulos que se comunicam pela troca de mensagens, esses podem ser classificados em módulos simples ou compostos. Os módulos simples, também chamados módulos ativos, são escritos em C++ usando as bibliotecas de classe de simulação. Esses podem ser agrupados, formando os módulos compostos.

A combinação de módulos simples e/ou módulos compostos aninhados hierarquicamente é denominada modelo. Um modelo representa a implementação de um sistema ou um protocolo que se deseja modelar (OMNET MANUAL, 2011).

As mensagens utilizadas na comunicação entre módulos podem ser enviadas diretamente a um destinatário final ou a um caminho pré-definido, por meio de pontos de comunicação dos módulos. Tais mensagens podem conter estruturas complexas de dados de forma a garantir que informações necessárias às camadas adjacentes sejam preenchidas corretamente (OMNET MANUAL, 2010; GUIMARÃES, 2008).

Os módulos simples e compostos são instâncias de tipos de módulos. Enquanto descrevendo o modelo, o usuário define os tipos de módulos, instâncias desses tipos de módulos servem como componentes para tipos de módulos mais complexos. O usuário cria os módulos do sistema como um módulo da rede que é um tipo especial de módulo composto sem portas para o mundo externo. Quando um tipo de módulo é usado para construção de blocos, não há a distinção se o mesmo é um módulo simples ou composto. Possibilitando ao usuário uma transparência na divisão de um módulo em vários módulos simples dentro de um módulo composto, ou reimplementando a funcionalidade de um módulo composto em um módulo simples, sem afetar a existência do uso dos tipos de módulos. No OMNeT++ a viabilidade do reuso é fornecida por modelos de frameworks, como os disponíveis na página oficial do simulador, dos quais tem-se:

- *INET Framework*: que contém modelos de implementação para os protocolos IP, TCP, UDP, PPP, Ethernet, MPLS e outros. E inclui suporte a simulações móveis e sem fio.
- MiXiM: que é a fusão de vários frameworks do OMNeT++, escrito para suportar simulações móveis e sem fio.
- xMIPv6: que é um modelo de simulação preciso e extensível do IPv6 móvel para o *INET Framework*.
- ReaSE: que é uma extensão do *INET Framework*, capaz de criar ambientes de simulação realísticos com relação a topologias de rede hierárquica, fornece uma interface gráfica para geração de arquivos NED, incluindo uma topologia realística e necessária para entidades na geração de tráfego.
- OverSim: que tem como objetivo executar uma série de requisitos parcialmente negligenciados pelos *frameworks* de simulação *peer-to-peer*. A primeira versão inclui implementações dos protocolos *peer-to-peer* de Chord e GIA.
- *Mobility Framework*: que suporta simulações de redes sem fio e móveis dentro do OMNeT++, como por exemplo redes *ad-hoc* e redes de sensores. O *Mobility Framework* será descrito em mais detalhes na Seção A.2.

A estrutura de um modelo de simulação no OMNeT++, é descrita utilizando a linguagem NED, que será apresentada na próxima seção.

A.1.2 Linguagem NED

A linguagem NED tem suporte a Descrição de Rede (NED *stands for Network Description*) e permite que os usuários declarem módulos simples e os agrupem em módulos compostos. O usuário pode classificar alguns módulos compostos como redes, os chamados modelos de simulação autônomos. Há também os canais que são outros tipos de componentes, cujas instâncias podem ser

usadas como módulos compostos (OMNET MANUAL, 2011; VARGA e HORNIG, 2008).

A linguagem NED possui várias características que permite que a mesma possa ser utilizada com sucesso em grandes projetos, tais características são elencadas a seguir:

- **Hierarquia**: no OMNet++ qualquer módulo que seja muito complexo como uma única entidade pode ser quebrado em módulos menores e usado como um módulo composto;
- **Baseada em Componentes**: módulos simples e módulos compostos são reutilizáveis, o que reduz a cópia de código e permite a existência de bibliotecas de componentes, como por exemplo, o *INET Framework* e o *MiXiM*;
- **Interfaces**: interfaces de módulos e canais podem ser usadas como um espaço reservado, onde normalmente um tipo de módulo ou canal seria utilizado;
- **Herança**: módulos ou canais podem ser subclasses. Os módulos e canais derivados podem adicionar novos parâmetros, portas e novos submódulos e conexões;
- **Pacotes**: a linguagem NED apresenta uma estrutura como um pacote Java, para reduzir o risco de confrontos entre diferentes modelos. NEDPATH, similar ao CLASSPATH do Java, foi introduzida para tornar mais fácil especificar dependências entre os modelos de simulação;
- **Tipos de entrada**: são os tipos de canais e os tipos de módulos usados localmente por um módulo composto;
- **Anotações de Metadados**: é possível anotar os tipos de módulos ou canais, parâmetros, portas e submódulos por meio da adição de propriedades. Metadados não são usados diretamente pelo núcleo da simulação, mas podem carregar informações extras para várias ferramentas, como o ambiente de execução ou mesmo para outros módulos no modelo.

A.2 *Mobility Framework*

O *Mobility Framework* (MF) foi desenvolvido como uma forma de inserir mobilidade ao OMNeT++ por vários pesquisadores do *Telecommunication Networks Group* na *Technische Universitaet Berlin*.

A primeira versão chamada FraSiMO, foi escrita por Heiko Sheunemann e Daniel M. Kirsch em 2001. Entre 2002 e 2003 essa versão foi revisada por Steffe Sroka, que reescreveu o *framework* completamente para melhorar a integração e aumentar a velocidade (FraSiMOII) com o OMNeT++.

Em 2003, Marc Loebbers iniciou o desenvolvimento da atual versão do MF, adicionando a manipulação de portas dinâmicas e iniciando a estruturação e documentação do código. Loebbers contou com uma grande ajuda de Daniel Willkomn, durante o processo de desenvolvimento do MF o *ChannelControl* e *Blackboard* foram completamente reescritos por Adreas Koepke.

O MF foi desenvolvido para suportar simulações sem fio e móveis dentro do OMNeT++. O núcleo desse *framework* implementa o suporte a mobilidade de nós, o gerenciamento de conexão dinâmica e um modelo de canal sem fio. Adicionalmente o *framework* também fornece ao usuário módulos básicos que podem ser utilizados e/ou derivados em ordem para implementar seus próprios módulos. Com este conceito um programador pode facilmente desenvolver suas implementações de protocolos para o MF sem ter que lidar com as interfaces e outras características necessárias a interoperabilidade. Segundo (LÖBBERS e WILLKOMM, 2010), o MF pode ser usado para simular:

- Redes fixas sem fio;
- Redes móveis sem fio;
- Redes centralizadas e distribuídas (*ad-hoc*);
- Redes de sensores;
- Multicanais para redes sem fio;
- E, muitas outras simulações que necessitem de suporte a mobilidade e/ou uma interface sem fio.

A.2.1 Conceitos básicos

Nesta seção são apresentados conceitos básicos que estão por trás do MF, sendo introduzidas as funções relevantes dos módulos *Basic**. A classe *BasicModule,* por exemplo, é comum a todas as classes dos submódulos *Host.* A *BasicModule* em si é derivada da *cSimplemodule* e da *BlackboardAccess* do OMNeT++, que fornece a funcionalidade de subscrever e publicar informação no módulo *Blackboard* (LÖBBERS e WILLKOMM, 2010).

Alguns dos conceitos básicos do MF são descritos a seguir:

- *BasicModule*: usa múltiplos estágios de inicialização do OMNeT++, assim todos os módulos no MF tem que ser implementados com dois estágios de inicialização.
- Conceito de endereço: usa o módulo *id()* do OMNeT++ para endereçamento no MF. O *nic* do módulo *id()* é usado como endereço Mac e o *netwLayer* é utilizado como endereço da camada de rede ou como endereço da camada de aplicação.
- Padrões de nomeação: existem padrões de nomeação para os módulos nos arquivos NED que devem ser seguidos, caso os mesmos não sejam seguidos ocorrerá falhas de segmentação na execução do código. Por exemplo, todos os módulos *hosts* têm que incluir os caracteres *host* ou *Host* em alguma parte de seu nome.
- Módulos *Basic**: são derivados do *BasicModule* e são definidos a cada camada para definir interfaces fáceis de entender e estender se necessário. A idéia é ter a possibilidade de facilmente estender ou adaptar os módulos de diferentes camadas aos requisitos específicos da simulação. Para este propósito existe as funções *handle*Msg()* e as funções *convenience*.
- Funções *Handle*Msg()*: contêm a funcionalidade do protocolo real, são chamadas cada vez que uma mensagem chega e contêm toda a informação necessária ao processamento e transmissão das mensagens. O MF fornece três tipos de funções para manipular três tipos de mensagens de eventos possíveis:

- *handleSelfMsg*: é o espaço reservado para manipular todas as questões relacionadas ao tempo e para iniciar ações em tempo de saída. *handleSelfMsg* é considerada a maneira mais fácil de implementar temporizadores no OMNeT++.
- *handleUpperMsg*: é chamada sempre que uma mensagem chega de uma camada superior.
- *handleLowerMsg*: para as mensagens das camadas mais baixas é o caminho inverso. Depois que o processamento é iniciado se necessário elas serão encaminhadas para camadas superiores. Isto é feito usando a função *sendUp()* que também é responsável pelo desencapsulamento.

- Funções *Convenience:* são definidas para facilitar interfaces comuns e ocultar inevitáveis interfaces de gerenciamento do usuário. O MF fornece algumas funções *convenience*, listadas a seguir:
  - *encapsMsg*: é chamada logo depois que uma mensagem chega das camadas superiores, é responsável por encapsular a mensagem da camada *n*+1 em uma mensagem a camada *n*.
  - *sendUp*: é chamada se uma mensagem for encaminhada para as camadas superiores, usualmente dentro das *handleLowerMsg()*. É responsável por desencapsular as mensagens antes de enviá-la para o módulo da camada *n*+1.
  - *sendDown*: envia mensagens para a camada *n*-1.
  - *sendDelayUp*: usada quando se deseja atrasar o ponto do tempo para enviar a mensagem a camada superior. O tempo que a mensagem deve ser atrasada pode ser dado como um parâmetro.
  - *sendDelayDown*: o mesmo que o *sendDelayUp*, mas para atrasar o envio de mensagens a camadas inferiores.

ANEXO B – Publicações

- EPOCA 2009 (2ª Escola Potiguar de Computação e suas Aplicações) – Abordagem com interação entre as camadas de aplicação, roteamento e MAC para redes de sensores sem fio;

- ITS 2010 (*7º International Telecommunications Symposium*) – Cross-layer interaction between the application, Routing and MAC Layer for Wireless Sensor Networks;

- MOMAG 2010 (14º Simpósio Brasileiro de Microondas e Optoeletrônica e 9º Congresso Brasileiro de Eletromagnetismo) – Impacto do relacionamento cruzado entre camadas sobre o desempenho de redes de sensores sem fio;

- JCC 2011 (*Journal of Communication and Computer*) – Cross-Layer Interaction among the Application, Routing and MAC Layers for Wireless Sensor Networks.