



**ALEXANDRE VICTOR FASSIO**

**UM PROTOCOLO DE MÚTIPLAS CAMADAS PARA  
REDES DE SENSORES SEM FIO COM BAIXO  
CONSUMO DE ENERGIA E LATÊNCIA REDUZIDA**

**Lavras  
Minas Gerais - Brasil  
2013**

**ALEXANDRE VICTOR FASSIO**

**UM PROTOCOLO DE MÚLTIPLAS CAMADAS PARA REDES DE  
SENSORES SEM FIO COM BAIXO CONSUMO DE ENERGIA E  
LATÊNCIA REDUZIDA**

Monografia apresentada ao Colegiado do  
Curso de Ciência da Computação, para  
obtenção do título de Bacharel em Ciên-  
cia da Computação

Orientador

Prof. João Carlos Giacomini

Co-Orientador

Prof. Tales Heimfarth

**Lavras**

**Minas Gerais - Brasil**

**2013**

**ALEXANDRE VICTOR FASSIO**

**UM PROTOCOLO DE MÚLTIPLAS CAMADAS PARA REDES DE  
SENSORES SEM FIO COM BAIXO CONSUMO DE ENERGIA E  
LATÊNCIA REDUZIDA**

Monografia apresentada ao Colegiado do  
Curso de Ciência da Computação, para  
obtenção do título de Bacharel em Ciên-  
cia da Computação

Aprovada em *15 de Abril de 2013*

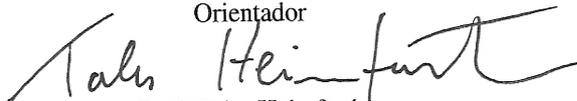
Prof. Raphael Winckler de Bettio

Prof. Luiz Henrique Andrade Correia



Prof. João Carlos Giacomin

Orientador



Prof. Tales Heimfarth

Co-Orientador

**Lavras**

**Minas Gerais - Brasil**

**2013**

**SUMÁRIO**

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>1</b>
<b>1.1</b>	<b>Redes de Sensores Sem Fio .....</b>	<b>1</b>
<b>1.2</b>	<b>Protocolos MAC .....</b>	<b>3</b>
<b>1.3</b>	<b>Objetivo .....</b>	<b>4</b>
<b>1.4</b>	<b>Estrutura do Trabalho .....</b>	<b>4</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO .....</b>	<b>6</b>
<b>2.1</b>	<b>Protocolos MAC (Enlace) .....</b>	<b>8</b>
<b>2.2</b>	<b>Protocolos MAC Baseados em Contenção para RSSF .....</b>	<b>11</b>
<b>2.2.1</b>	<b>S-MAC .....</b>	<b>12</b>
<b>2.2.2</b>	<b>B-MAC .....</b>	<b>13</b>
<b>2.2.3</b>	<b>X-MAC .....</b>	<b>15</b>
<b>2.3</b>	<b>Protocolos de Roteamento em Redes de Sensores Sem Fio.....</b>	<b>16</b>
<b>2.3.1</b>	<b>Protocolo LEACH.....</b>	<b>18</b>
<b>2.3.2</b>	<b>Protocolo Difusão Direta .....</b>	<b>20</b>
<b>2.3.3</b>	<b>Protocolo GEAR .....</b>	<b>22</b>
<b>2.3.4</b>	<b>Protocolo PROC .....</b>	<b>24</b>
<b>3</b>	<b>METODOLOGIA .....</b>	<b>27</b>
<b>3.1</b>	<b>Formação do <i>Backbone</i> .....</b>	<b>36</b>
<b>4</b>	<b>RESULTADOS .....</b>	<b>39</b>
<b>5</b>	<b>CONCLUSÃO.....</b>	<b>50</b>
<b>5.1</b>	<b>Trabalhos Futuros .....</b>	<b>51</b>

## LISTA DE FIGURAS

Figura 1	Camadas Definidas no Modelo IEEE802.15.4. Fonte: adaptado de IEEE (2006).....	7
Figura 2	Problema da estação oculta. Fonte: adaptado de (TANENBAUM, 2003). ....	11
Figura 3	Relação de tempos entre o transmissor e o receptor no protocolo S-MAC. Fonte: adaptado de Ye, Heidemann e Estrin (2004).....	13
Figura 4	Comparação entre IEEE802.15.4, B-MAC (LPL) e X-MAC. Fonte: adaptada de (BUETTNER <i>et al.</i> , 2006).....	16
Figura 5	Aprender rotas em torno de "buracos" (YU; GOVINDAN; ESTRIN, 2001). ....	23
Figura 6	Difusão de dados em uma região utilizando encaminhamento geográfico recursivo (YU; GOVINDAN; ESTRIN, 2001). ....	24
Figura 7	Processo de criação do <i>backbone</i> no protocolo PROC. Fonte: adaptado de Macedo <i>et al.</i> (2006) .....	26
Figura 8	RSSF utilizando zonas de tráfego rápido.....	29
Figura 9	Exemplo de inclusão aleatória de um nó comum ao <i>backbone</i> .....	30
Figura 10	Exemplo de um <i>backbone</i> formado sem otimizações.....	32
Figura 11	Exemplo de um <i>backbone</i> formado com otimizações. ....	32
Figura 12	Exemplo de inclusão de um nó comum ao <i>backbone</i> de acordo com a Distância Máxima. ....	33
Figura 13	Exemplo de inclusão desnecessária de nós sensores ao <i>backbone</i> . .	36
Figura 14	Exemplo de uma rede e <i>Backbone</i> formados sem otimizações.....	39
Figura 15	Exemplo de uma rede e <i>Backbone</i> formados com otimizações. ....	40
Figura 16	Média de pacotes transmitidos para formação do <i>backbone</i> .....	42
Figura 17	Média de falhas para formar o <i>backbone</i> .....	43
Figura 18	Número médio de nós necessários para formar o <i>backbone</i> .....	44
Figura 19	Comprimento médio do <i>backbone</i> .....	45

Figura 20	Distância média de cada nó até a base .....	46
Figura 21	Distância média de cada nó comum até o <i>backbone</i> .....	48

**LISTA DE TABELAS**

Tabela 1	Média e Desvio Padrão (DP) para número de pacotes transmitidos para formação do <i>backbone</i> . ....	42
Tabela 2	Média e Desvio Padrão (DP) para número de nós necessários para formar o <i>backbone</i> .....	44
Tabela 3	Média e Desvio Padrão (DP) para comprimento do <i>backbone</i> . ....	46
Tabela 4	Média e Desvio Padrão (DP) para distância de cada nó até a base..	47
Tabela 5	Média e Desvio Padrão (DP) para distância de cada nó comum até o <i>backbone</i> . ....	48

## Resumo

Em Redes de Sensores sem Fio (RSSF), a eficiência em energia é um dos requisitos mais importantes. Assim, o mecanismo de ciclos de trabalho é utilizado nos protocolos de controle e acesso ao meio (MAC) em RSSF, pois, permite uma redução no consumo de energia. Nesse mecanismo os nós sensores frequentemente desligam seus rádios para economizarem energia e liga-os por um curto período para receberem ou transmitirem alguma informação. Entretanto, o uso de ciclos de trabalho aumenta a latência (atraso nas transmissões) da rede. Este trabalho propõe um novo protocolo que envolve aspectos de roteamento e controle de acesso ao meio. O objetivo deste trabalho é reduzir o consumo de energia e latência de uma RSSF quando utiliza-se ciclos de trabalho. Isso será possível através da criação de uma rota de tráfego rápido (backbone). Os nós sensores que compõem essa rota serão responsáveis por transmitirem as mensagens até a base. Essa transmissão será livre de latência porque utiliza um alto ciclo de trabalho. Enquanto que os demais nós permanecerão em baixo ciclo de trabalho.

Palavras-chave: Rede de Sensores sem Fio, protocolo de controle e acesso ao meio, RSSF, MAC, simulação de redes.

# 1 INTRODUÇÃO

## 1.1 Redes de Sensores Sem Fio

Uma Rede de Sensores Sem Fio (RSSF) é um caso especial de rede de comunicação de dados composta por um grande número de nós (os nós sensores), os quais se comunicam por múltiplos saltos (*multi-hop*) através de minúsculos rádios e são equipados com processadores embarcados, memória e sensores (HEIDEMANN; GOVINDAN, 2004). Em alguns casos a comunicação é feita por sinais óticos (luz ou laser). O objetivo de uma rede de sensores é medir (sensoriar) e coletar dados do ambiente em pontos distintos de uma região e transmitir esses dados até um ponto de coleta (ponto de acesso).

Porém, nem sempre a implantação direta de uma RSSF é viável, pois, a incerteza quanto ao funcionamento dela pode não gerar os resultados esperados. Pode acontecer também, que um mal planejamento da topologia da rede gere interferências nos rádios e mal funcionamento da RSSF. Por isso, devem ser feitas simulações em computador para testar algoritmos de RSSF antes de colocá-las em campo.

As Redes de Sensores Sem Fio (RSSF) possuem custo baixo de implantação e manutenção (TAVARES; VELEZ; FERRO, 2008; JANEK; EVANS, 2010), e podem ser aplicadas em diversas áreas: científica, militar, comercial, ambiental, etc (LOUREIRO *et al.*, 2003; TAVARES; VELEZ; FERRO, 2008; JANEK; EVANS, 2010). As aplicações podem ser enquadradas em uma das seguintes classes: detecção de eventos, monitoramento de ambientes com envio periódico de dados, monitoramento com requisição de dados pelo nó-base, aplicações de identificação e rastreamento de alvos (IYER *et al.*, 2008). Este trabalho se destina a aplicações

da primeira e segunda classe, nas quais as comunicações ocorrem quase exclusivamente no sentido dos nós sensores para o ponto de acesso (*sink*). Na primeira classe, os dados são enviados apenas quando um determinado evento de interesse ocorrer, como por exemplo, o vazamento indevido de gás. Na segunda classe, os dados são enviados periodicamente dos nós sensores para o nó-base.

As RSSF tem sido alvo de pesquisa nos últimos 10 anos (ALI *et al.*, 2012) e muitos de seus aspectos, embora bastante estudados ainda permanecem com questões abertas para pesquisa. Uma dessas questões é a conjugação dos dois requisitos mais importantes para o funcionamento dessas redes, quais sejam o baixo consumo de energia pelos nós sensores e a baixa latência na transmissão de dados.

Neste trabalho foi desenvolvido um algoritmo de roteamento e controle de acesso ao meio de comunicação (MAC - *Medium Access Control*) de uma Rede de Sensores Sem Fio. Este algoritmo foi desenvolvido para aplicações em que o tráfego de informações ocorre principalmente no sentido da rede para o nó base (*sink*). Este algoritmo terá como principais metas a eficiência em energia e a baixa latência.

Para uma RSSF, um projeto eficiente em energia é considerado como prioridade máxima (YE; HEIDEMANN; ESTRIN, 2004). Seus requisitos principais são:

- Eficiência em energia (YE; HEIDEMANN; ESTRIN, 2004; POLASTRE; HILL; CULLER, 2004; YICK; MUKHERJEE; GHOSAL, 2008) - os nós sensores são alimentados por baterias e a troca ou recarga destas é inviável.
- Escalabilidade (YE; HEIDEMANN; ESTRIN, 2004; POLASTRE; HILL; CULLER, 2004; YICK; MUKHERJEE; GHOSAL, 2008) - uma rede de

sensores deve conter um grande número de nós sensores e estes podem se mover, podem sair de operação ou novos nós podem ser adicionados à rede.

- Latência (YICK; MUKHERJEE; GHOSAL, 2008; NOROUZI; ZAIM, 2012) - dependendo da aplicação, algum atraso de propagação de mensagens pode ser tolerado. A latência e a conservação de energia são requisitos conflitantes no projeto de protocolos MAC de redes sem fios.
- Justiça (YICK; MUKHERJEE; GHOSAL, 2008) - todos nós sensores devem ter chances para realizar transmissões. Mas, normalmente, justiça não é um aspecto crítico em RSSF, desde que todos os nós servem a uma mesma aplicação.
- Utilização eficiente da faixa de frequências de comunicação de rádio (YICK; MUKHERJEE; GHOSAL, 2008).

## 1.2 Protocolos MAC

Os protocolos de controle e acesso ao meio (MAC - Medium Access Control) podem ser divididos em duas categorias: baseados em escala (reserva de recursos) e baseados em contenção (YE; HEIDEMANN; ESTRIN, 2004). Neste trabalho foi desenvolvido um protocolo baseado em contenção, pois, este tipo de protocolo permite uma boa escalabilidade e é mais simples (YE; HEIDEMANN; ESTRIN, 2004). Para propósito de eficiência energética os nós sensores utilizarão ciclos de trabalho (*duty-cycle*), permanecendo em operação em uma pequena parte do tempo e ficando desligados no restante do tempo (BUETTNER *et al.*, 2006). Tal ciclo ocorre em intervalos periódicos. A adoção de ciclos de trabalho causa um aumento na latência (LI; YE; HEIDEMANN, 2005), ou seja, as mensagens demo-

ram mais tempo para chegar ao *sink* (nó base). Esta é uma situação indesejável principalmente em aplicações de detecção de eventos (primeira classe).

O problema a ser tratado é a redução da latência inerente nas transmissões das RSSF quando se utilizam mecanismos de redução de energia, como a adoção de ciclos de trabalho (*duty-cycle*). Este é um problema que tem sido pesquisado em RSSF nos últimos dez anos, mas poucas propostas abordando os dois aspectos têm sido apresentadas. Para reduzir a latência é proposto um algoritmo de acesso ao meio que envolve aspectos de roteamento, o que é denominado protocolo de múltiplos níveis (*cross-layer*) (SRIVASTAVA; MOTANI, 2005). Neste protocolo alguns nós da rede permanecerão ligados a maior parte do tempo criando uma rota de alta velocidade em direção ao *sink*, enquanto os demais nós da rede adotam ciclos de trabalho para reduzir o consumo de energia.

### **1.3 Objetivo**

O objetivo deste trabalho foi desenvolver um protocolo de múltiplas camadas (*cross-layer*) para Redes de Sensores Sem Fio (RSSF), envolvendo roteamento e controle de acesso ao meio, funções desempenhadas pela Camada de Rede e pela Camada de Enlace, respectivamente, com o intuito de se alcançar baixa latência e alta eficiência energética.

### **1.4 Estrutura do Trabalho**

Este trabalho está organizado da seguinte forma:

O Capítulo 2 apresenta o referencial teórico, abordando aspectos sobre Redes de Sensores Sem Fio, tais como as pesquisas científicas envolvendo protocolos

de comunicação e objetivos desses protocolos, sendo que alguns objetivos desses são o roteamento eficiente de mensagens pela rede ou economia de energia dos nós sensores. O Capítulo 3 mostra a metodologia de trabalho utilizada para desenvolver o protocolo de múltiplas camadas proposto. O Capítulo 4 apresenta os resultados obtidos. E o Capítulo 5 apresenta as conclusões.

## 2 REFERENCIAL TEÓRICO

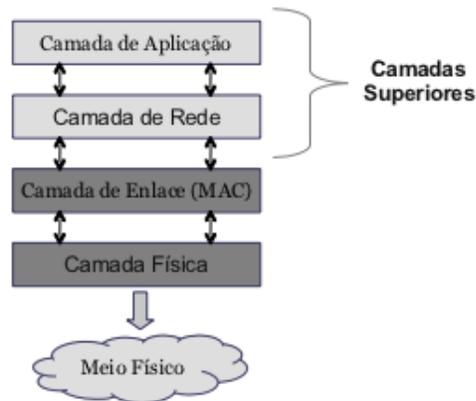
As Redes de Sensores Sem Fio (RSSF) foram objeto de estudo durante as últimas décadas, diversas pesquisas foram feitas, sobretudo para melhorar a economia de energia nessas redes (ALI *et al.*, 2012). Segundo Loureiro *et al.* (2003) e Torre *et al.* (2012), a criação das RSSF foi possível devido ao grande avanço tecnológico nas áreas de sensores, circuitos integrados e comunicação sem fio. Além disso, Torre *et al.* (2012) ressaltam que o desenvolvimento de dispositivos com baixo consumo de energia tornaram as RSSF uma das soluções mais versáteis e dessa forma é possível obter uma grande gama de aplicações.

Portanto, esse tipo de rede vem ganhando atenção mundial devido à sua importância e aplicabilidade, de modo que os nós sensores de uma RSSF podem ser usados para perceber, medir e transmitir informações do ambiente e posteriormente transmitir os dados obtidos ao usuário (YICK; MUKHERJEE; GHOSAL, 2008). Assim, as RSSF “fornecem uma forma de monitorar um ambiente com maior granularidade espacial e temporal” (JANEK; EVANS, 2010).

Segundo Janek e Evans (2010), as RSSF são usadas para uma variedade de aplicações, tais como, monitoramentos de segurança, controle de tráfego, monitoramento de vida selvagem ou monitoramento de fábricas. Elas podem ser úteis em ambientes perigosos, impróprios para a presença humana, como por exemplo, um campo de base inimiga ou um campo tóxico de materiais radioativos.

As RSSF compõem uma parcela particular das redes de comunicação de dados, as quais são geograficamente restritas, exigem pouco processamento de dados e, na maioria das vezes, são compostas por equipamentos compatíveis entre si, o que as enquadram em nova categoria, PANs (*Personal Area Networks*) (IEEE,

2006). Estas características permitem que sejam feitas muitas simplificações em seus protocolos de controle. Além disso, algumas exigências próprias das RSSF, como o acesso a algumas funções de hardware e economia de energia nas comunicações, não permitem que protocolos de redes existentes sejam utilizados (HILL *et al.*, 2004). Dessa forma, a IEEE (2006) indica quatro camadas para os protocolos de comunicação para uma RSSF baseado no modelo ISO/OSI, sendo: camada de Aplicação, Rede, Enlace (MAC) e Física. Porém, na especificações desse padrão – padrão IEEE802.15.4 (IEEE, 2006) – para RSSF define-se somente as camadas de Enlace e Física. A figura 1 ilustra o protocolo IEEE802.15.4 para redes de sensores sem fio. Em geral, os protocolos de comunicação são definidos para que haja um padrão a ser seguido de forma que a comunicação entre dois dispositivos possa ser feita de maneira correta.



**Figura 1:** Camadas Definidas no Modelo IEEE802.15.4. Fonte: adaptado de IEEE (2006).

## 2.1 Protocolos MAC (Enlace)

De maneira geral os protocolos MAC para Redes de Sensores Sem Fio podem ser divididos em duas categorias conforme Ye, Heidemann e Estrin (2004): baseados em escala (reserva de recursos) e baseados em contenção.

Protocolos MAC baseados em escala funcionam com a ideia de reserva e agendamento do intervalo para que uma transmissão seja realizada. Basicamente, essa categoria é do tipo TDMA (*Time Division Multiple Access*), de modo que há um nó centralizador (mestre) que distribui a escala de transmissão entre os nós da rede durante a inicialização. Para utilizar este tipo de protocolo, uma RSSF é normalmente dividida em *clusters*, e o líder (*clusterhead*) controla o tempo de acesso de cada nó de seu grupo.

Há também outros dois tipos de protocolos baseados em reserva de recursos: CDMA (*Code Division Multiple Access*) e FDMA (*Frequency Division Multiple Access*). Entretanto, TDMA é o tipo de protocolo baseado em escala mais utilizado por exigir um hardware não tão complexo. Essa categoria possui uma vantagem natural de economia de energia quando comparado aos protocolos baseados em contenção, devido ao fato de que o ciclo de trabalho é reduzido e não há *overhead* e colisões entre pacotes transmitidos na rede. Isso acontece pelo fato de que o nó mestre realiza o agendamento das transmissões, ou seja, um nó que deseje transmitir fará em seu tempo determinado pelo nó mestre, evitando que dois nós disputem o meio para realizar algum envio de mensagem.

Ye, Heidemann e Estrin (2004) ressaltam, porém, algumas dificuldades e desvantagens encontradas nos protocolos baseados em escala: não possuem uma boa escalabilidade, isto é, adaptar-se a uma alteração na topologia da rede ou aumen-

to/redução do número de nós sensores é um fator difícil de ser tratado por esses protocolos; geralmente é necessário a formação de *clusters* de comunicação com os nós sensores como acontece no protocolo LEACH (HEINZELMAN; CHANDRAKASAN; BALAKRISHNAN, 2000). O gerenciamento de interferências e de comunicação entre *clusters* não são tarefas triviais. Além disso, muitos nós sensores estão restritos para realizar a comunicação apenas com os nós pertencentes ao mesmo *cluster* (YE; HEIDEMANN; ESTRIN, 2004).

Protocolos MAC baseados em contenção são amplamente utilizados em redes de sensores sem fios por causa de sua simplicidade (YE; HEIDEMANN; ESTRIN, 2004). Basicamente os protocolos baseados em contenção são do tipo CSMA (*Carrier Sense Multiple Access*). Nesse tipo de protocolo, quando um nó deseja transmitir alguma mensagem, este deverá primeiramente ligar o receptor e verificar se já há alguma comunicação sendo feita. Caso seja detectado que há uma comunicação ocorrendo no momento da verificação, o nó que deseja transmitir deverá adiar sua transmissão.

Um dos problemas envolvendo este método são as escutas ociosas, isto é, o rádio fica operando como receptor, esperando que uma mensagem seja enviada para ele. Mas, em uma RSSF, as mensagens são raras, portanto, o rádio poderia ficar a maior parte do tempo desligado para economizar energia. Então, devem ser criados modos de controle de operação para que o rádio esteja ligado na hora que houver alguma transmissão para ele. Segundo Polastre, Hill e Culler (2004) as escutas ociosas são as maiores causadoras de desperdício de energia em uma rede de sensores sem fio.

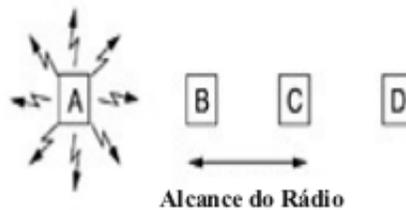
Outro problema corrente neste tipo de protocolo são as colisões, que ocorrem por não haver um nó centralizador que administra e agenda as transmissões, logo, haverá frequentes disputas para a realização de transmissões.

De maneira geral, as redes sem fio utilizam a técnica de CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) que evita a ocorrência de colisões (TANENBAUM, 2003). Nessa técnica, antes de uma estação transmitir alguma mensagem, ela deverá escutar o meio para verificar se já não existe nenhuma transmissão ocorrendo. Se o meio estiver ocupado, deverá aguardar até que o meio esteja disponível para transmissões.

O principal causador de colisões é o problema da estação oculta, conhecido na literatura como *hidden terminal problem*. Por exemplo, considere o que acontece quando a estação A está transmitindo para a estação B, como mostra a Figura 2. Quando a estação C verificar o meio físico, não ouvirá a estação A, pois, essa ela está fora de alcance. Portanto, a estação C concluirá incorretamente que pode efetuar a transmissão para a estação B. Então, se as estações A e C transmitirem simultaneamente, C impedirá que B receba a mensagem de A. Neste caso, haverá colisão. O problema de uma estação não conseguir detectar uma provável concorrente pelo meio físico, porque ambas estações estão fora do alcance entre si, é denominado problema da estação oculta (TANENBAUM, 2003).

Para tratar o problema da estação oculta, o padrão IEEE 802.11 (IEEE, 1999), que é um exemplo de um protocolo baseado em contenção, aplica um mecanismo opcional de RTS/CTS (pacotes de controle).

Apesar desta desvantagem, os protocolos baseados em contenção possuem boa escalabilidade quando comparados aos protocolos baseados em escala, além de serem mais simples (YE; HEIDEMANN; ESTRIN, 2004).



**Figura 2:** Problema da estação oculta. Fonte: adaptado de (TANENBAUM, 2003).

## 2.2 Protocolos MAC Baseados em Contenção para RSSF

Diversos protocolos de controle de acesso ao meio (MAC) foram propostos com o objetivo de atingir certos requisitos importantes para o funcionamento eficiente de uma RSSF. A maior parte deles são baseados em contenção devido ao fato de facilitarem a escalabilidade da rede.

Polastre, Hill e Culler (2004) e Ye, Heidemann e Estrin (2004) identificam os seguintes requisitos como importantes para um protocolo MAC: baixo consumo de energia, prevenção efetiva de colisões, escalabilidade, uso eficiente do canal para transmissões. Além disso, Ye, Heidemann e Estrin (2004) propõe que atributos como justiça, latência, rendimento e utilização da largura de banda podem ser objetivos secundários.

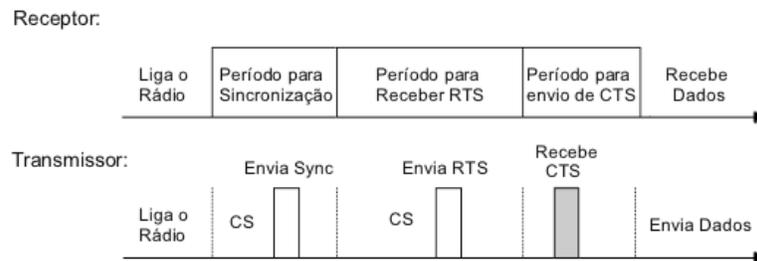
Por outro lado, eficiência energética é um dos temas mais abordados ao se projetar protocolos de comunicação para RSSF. Um dos mecanismos primordiais para atingir este requisito é o ciclo de trabalho (*duty cycle*). Nessa abordagem os nós permanecem em operação em uma pequena parte do tempo, ficando desligados no restante do tempo (BUETTNER *et al.*, 2006).

### 2.2.1 S-MAC

O protocolo S-MAC, proposto por Ye, Heidemann e Estrin (2004), utiliza a abordagem de ciclos de trabalho. Nessa técnica os nós periodicamente dormem (desligam seus rádios), acordam para eventualmente receber ou transmitir mensagens e voltam a dormir. Dessa forma evita-se o desperdício de energia, pois, o uso de ciclos de trabalho reduz a escuta ociosa, isto é, quando os nós sensores mantêm os rádios receptores ligados nos períodos em que não ocorrem transmissões na rede.

No protocolo S-MAC o período de sono de cada nó é definido através da sincronização com os nós vizinhos. Isso acontece no período para sincronizações, onde os nós trocam mensagens *SYNC* com seus vizinhos. Nessa abordagem, também é permitido que dois nós vizinhos possuam ciclos de trabalho diferentes, nesse caso, os nós terão seus ciclos de trabalho sincronizado com vizinhanças diferentes.

Para evitar colisões, no protocolo S-MAC utiliza-se o mecanismo CSMA/CA e troca de mensagens RTS/CTS. Dessa forma, a troca de mensagens RTS/CTS é importante para evitar o problema da estação oculta. Além disso, nesse protocolo, para transmissões *unicast* (transmissões entre apenas dois nós sensores), transmite-se a seguinte sequência de mensagens: RTS/CTS/DATA/ACK. A Figura 3 ilustra o tempo em que ocorrem as transmissões e o momento onde os nós sensores ligam e desligam seus rádios utilizando-se o protocolo descrito.



**Figura 3:** Relação de tempos entre o transmissor e o receptor no protocolo S-MAC. Fonte: adaptado de Ye, Heidemann e Estrin (2004)

### 2.2.2 B-MAC

O protocolo B-MAC proposto por Polastre, Hill e Culler (2004) teve como objetivo alcançar uma implementação mais simples, de código e ocupação de memória pequena. Portanto, diferente do protocolo S-MAC, B-MAC deixa os serviços de rede, tais como sincronização e roteamento, para serem realizados em uma camada acima.

Outra diferença é que nesse protocolo, o mecanismo utilizado para evitar colisões é o CCA (*Clear Channel Assessment*), evitando assim o uso de pacotes RTS e CTS durante a transmissão de dados. A técnica de CCA permite que um nó sensor verifique o canal antes de realizar qualquer transmissão. Caso o canal esteja ocupado, o nó tentará transmitir a mensagem novamente após um tempo.

Além disso, o protocolo B-MAC adota uma abordagem diferente, denominada *Low Power Listening (LPL)*, com a finalidade de evitar o desperdício de energia. Nessa abordagem, os nós sensores acordam por um curto período e verificam se há atividade no canal. Se não há nenhuma atividade (transmissões) o nó desligará seus rádios novamente. Caso contrário, o nó permanecerá acordado o tempo necessário

para receber a mensagem. Finalizada a recepção da mensagem, o nó voltará ao seu estado de sono, isto é, quando o nó encontra-se com seus rádios desligados.

Já o transmissor, quando desejar transmitir alguma mensagem, primeiramente, ele deverá enviar um preâmbulo longo para assegurar que o receptor receberá a mensagem de dados. Através do envio de um preâmbulo longo, garante-se que o nó receptor estará acordado para receber a mensagem de dados. Evitando, assim, o uso de sincronização entre os nós sensores. A escolha do tamanho do preâmbulo contribui para a aumentar a confiança da entrega das mensagens e também contribui para o aumento ou para a redução do consumo de energia. Portanto, nesse protocolo, permite-se ajustar o tamanho dos preâmbulos de acordo com as condições da rede.

Porém, um problema dessa abordagem é aumentar o *overhearing* já que no envio de preâmbulos não há qualquer informação sobre qual nó sensor será o receptor. Assim, todos os nós que escutam a transmissão do preâmbulo deverão aguardar até o final do envio do preâmbulo. Após o envio do preâmbulo, o transmissor enviará a mensagem de dados contendo o endereço do receptor. Portanto, agora somente o nó receptor permanecerá acordado para receber a mensagem. e o restante dos nós que ouviram o envio do preâmbulo, agora, poderão desligar seus rádios para economizarem energia. No período em que os nós sensores permanecem à espera do final do envio do preâmbulo, há desperdício de energia por parte dos nós receptores porque estes precisam esperar o final da transmissão para descobrirem a quem se destina a mensagem de dados.

Uma técnica semelhante de preâmbulo longo foi utilizada por El-Hoiydi (2002). O protocolo proposto por ele foi o *Aloha* com preâmbulo, em que o objetivo principal foi reduzir a escuta ociosa, ou seja, haverá uma redução no desperdício de

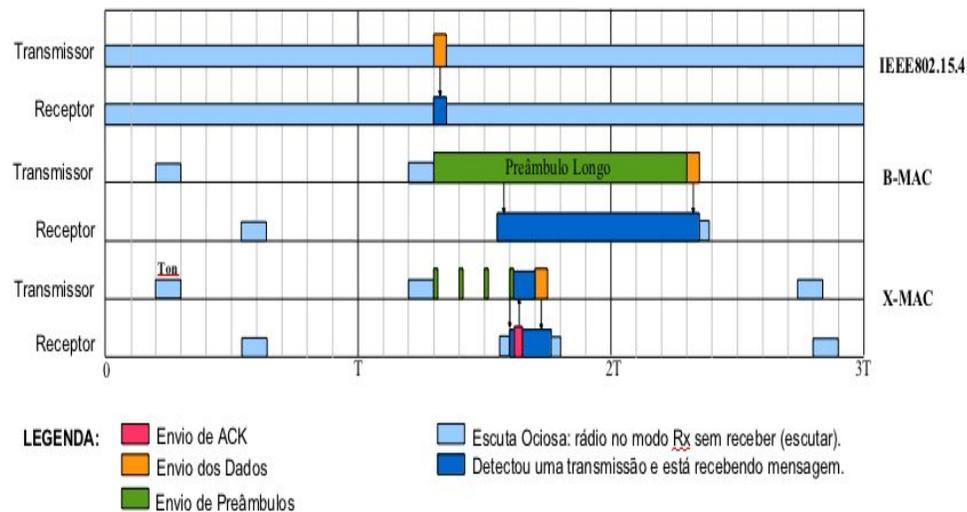
energia. Porém, como identificado pelo próprio autor, essa técnica aumenta o tempo de transmissão e a probabilidade de ocorrer colisões.

### 2.2.3 X-MAC

Assumindo que a abordagem que utiliza preâmbulos longos traz algumas desvantagens, como o aumento da latência, Buettner *et al.* (2006) desenvolveram um novo protocolo, chamado X-MAC. Nesse protocolo utiliza-se preâmbulos curtos com o objetivo de reduzir o consumo de energia e a latência. A primeira ideia incorporada ao X-MAC foi adicionar ao preâmbulo informações sobre o endereço do destinatário. Dessa forma, reduz-se o *overhearing* por parte dos nós não-alvo.

A segunda ideia incorporada ao X-MAC, permite que os nós sensores enviem preâmbulos curtos intercalado de pausas. Assim, quando um nó destinatário recebe o preâmbulo, ele poderá informar ao transmissor que já está pronto (rádios ligados) para receber a mensagem de dados – isto é feito a partir do envio de uma mensagem de reconhecimento (ACK) durante as pausas. Por sua vez, o transmissor ao receber uma mensagem de reconhecimento, interromperá a transmissão de preâmbulos e dará início ao envio da mensagem de dados. Entretanto, o uso de envio de preâmbulos curtos dificulta as transmissões em *broadcasting*.

Através da Figura 4, adaptada de Buettner *et al.* (2006), ressalta-se a vantagem de se utilizar o protocolo proposto por eles. Pode-se notar pela Figura 4 que o tempo gasto para a realização de uma transmissão é bem menor no protocolo X-MAC em relação ao B-MAC para o caso teste proposto pelos autores. Dessa forma, eles demonstram a importância da economia de energia e diminuição de latência ao se utilizar o protocolo X-MAC. Além disso, foi incluída nessa figura (Figura 4) o funcionamento do protocolo IEEE802.15.4 (IEEE, 2006), em que os



**Figura 4:** Comparação entre IEEE802.15.4, B-MAC (LPL) e X-MAC. Fonte: adaptada de (BUETTNER *et al.*, 2006)

nós sensores permanecem o todo tempo com seus rádios ligados. Portanto, o gasto de energia desse protocolo será maior do que os outros dois protocolos comparados. O mecanismo de ciclos de trabalho, porém, é sugerido pelos autores desse protocolo como uma forma de reduzir o consumo de energia.

### 2.3 Protocolos de Roteamento em Redes de Sensores Sem Fio

Frequentemente, uma questão fundamental de projeto para uma rede de comunicação é determinar a maneira como os pacotes são roteados da origem até o destino, ou seja, os protocolos de roteamento encarregam-se da transmissão fim a fim (TANENBAUM, 2003).

Em uma RSSF os requisitos para protocolos de roteamento dependem da aplicação onde será implantada a rede. Por exemplo, os requerimentos de um protocolo desenvolvido para aplicações ambientais são diferentes dos requerimentos

para uma aplicação militar (NOROUZI; ZAIM, 2012). Segundo os autores, alguns requerimentos que podem ser citados são: escalabilidade, latência e rendimento. Entretanto, eles ressaltam que independente dos requerimentos da aplicação, os protocolos de roteamento para RSSF devem buscar maximizar o tempo de vida da rede e minimizar o consumo de energia.

Além disso, segundo (AL-KARAKI; KAMAL, 2004) os protocolos de roteamento podem ser classificados de acordo com a estrutura da rede, podendo estes ser classificados como: protocolos de roteamento plano, hierárquico ou baseado em localização (roteamento geográfico).

Nos protocolos de roteamento plano, os nós sensores são ditos iguais por desempenharem a mesma função e colaboram entre si para realizarem uma tarefa comum. Enquanto que em roteamentos hierárquicos o objetivo é a formação de *clusters* e escolha de um líder (*cluster-head*) para cada *cluster*. Dessa forma, os líderes são encarregados de realizarem a agregação dos dados dos outros nós sensores para posteriormente transmitir os dados aglomerados até à base (*sink*). Portanto, nesse tipo de protocolo existem duas classes diferentes de nós sensores, diferentemente dos protocolos de roteamento plano em que todos nós possuem igual classificação.

Por sua vez, roteamento geográfico utiliza informações de coordenadas geográficas e distâncias entre os nós sensores para realizar a transmissão de dados pela rede. Muitas vezes, isso pode ser feito, por exemplo, aplicando-se um receptor de GPS (*Global Positioning System*) em cada nó sensor, obtendo assim, informações de coordenadas geográficas de cada nó sensor via satélite.

### 2.3.1 Protocolo LEACH

O protocolo LEACH (*Low-Energy Adaptive Clustering Hierarchy*) proposto por Heinzelman, Chandrakasan e Balakrishnan (2000) é um dos protocolos de roteamento frequentemente utilizados em RSSF quando a rede é homogênea, isto é, todos nós sensores têm a mesma configuração e capacidade.

Os autores propuseram um protocolo que utiliza a formação de grupos (*clusters*) através de ciclos, onde um nó sensor decide ser líder sem consultar os demais. Esse nó sorteia um número e compara com um valor limiar, se for menor ele se anuncia como líder. O valor limiar depende do número de *cluster-heads* (base local) determinados para aquela rodada, do número de vezes que esse nó sensor já foi líder e da quantidade de energia residual que ele possui. Posteriormente, o líder envia uma mensagem em *broadcast* para os demais, anunciando sua condição de líder. Os nós comuns escolhem a qual líder desejam se associar (menor custo de comunicação).

Após a formação dos *clusters*, o líder cria uma escala para os nós comuns poderem se comunicar, baseado em TDMA. Assim, segundo Norouzi e Zaim (2012) se este método de escolha fosse estático e não randômico, por fim o protocolo inutilizaria os nós sensores pertencentes aos *cluster-heads* muito rapidamente, já que o consumo de energia nesses será maior que nos demais nós sensores.

Além de definir a escala de tempo para transmissões, o (*cluster-head*) também coordena a atividade do grupo utilizando uma arquitetura de aglomeração, de forma que quando todos os nós membros tiverem enviado suas mensagens, o líder coloca todos os dados em uma única mensagem e envia ao nó base. Portanto, o protocolo LEACH também é classificado como protocolo de roteamento hierár-

quico devido ao fato de que alguns nós assumem a função de liderança sobre os outros nós sensores.

Norouzi e Zaim (2012) ressaltam algumas vantagens decorrentes das características do protocolo LEACH:

- Escalabilidade: há limites para a maioria das comunicações interiores aos *clusters*;
- Limite de Tráfego: os *cluster-head* agregam os dados coletados dos demais nós sensores, dessa forma, uma rede de grande escala pode ser desenvolvida sem que haja sobrecarga no tráfego da rede e ainda permite a redução do consumo de energia;
- Transmissões com um salto do nó sensor para o *cluster-head*. Logo, haverá uma maior economia de energia na rede;
- Aumento do tempo de vida da rede:
  1. Distribuição das funções de *cluster-head* para outros nós;
  2. Agregação de dados pelo *cluster-head*;
  3. A técnica de TDMA permite que muitos nós permaneçam em dormência quando não é seu momento de transmissão.

Em contrapartida, os autores também apontam os pontos de desvantagem ao se utilizar o protocolo LEACH:

- A rede em si possui alta dependência aos *cluster-heads*, sendo que problemas de robustez, tais como falhas nesses nós sensores trará grande prejuízo à rede;

- O protocolo LEACH não funciona bem em aplicações que cobrem grandes áreas e que requeiram comunicação multi-saltos entre *clusters*;

### 2.3.2 Protocolo Difusão Direta

O protocolo Difusão Direta (*Directed Diffusion*) proposto por Intanagonwiwat, Govindan e Estrin (2000) é um exemplo de um protocolo de roteamento plano e que também pode ser classificado como protocolo centrado em dados, pois, há a coleta e publicação de dados pela rede.

Com a finalidade de reduzir o consumo de energia, foi proposto que as interações entre os nós sensores sejam por troca de mensagens. Além disso, as requisições de informações são feitas pela base (*sink*), ou seja, ela envia uma requisição a fim de identificar qual nó sensor possui a informação necessária à base. Portanto, nesse protocolo a estação base transmite questões/perguntas em forma de requisições.

Por exemplo, o *sink* poderia transmitir uma mensagem aos nós sensores requisitando a seguinte informação: “quantos nós vizinhos há em sua região?”. Durante as transmissões de requisições via *broadcasting*, diversos nós sensores receberão a requisição encaminhada e poderão armazená-las em sua *cache* e posteriormente, os nós que receberam a requisição também a enviam para outros nós, da mesma forma como acontece com as transmissões da base. Um sensor, ao receber a requisição, ativa seu sensor para tentar obter as informações necessárias para posteriormente transmitir a informação para a base. Logo que possível, o nó que obtiver com sucesso a resposta para a requisição da base, poderia enviar-lhe toda a informação obtida.

De acordo com Norouzi e Zaim (2012) as vantagens de se utilizar o protocolo Difusão Direta são:

- O protocolo é utilizado para recuperar informações agregadas em um único nó;
- Ao contrário de outros protocolos de roteamento, há a possibilidade de mais de um *sink* realizar consultas e receber dados ao mesmo tempo;
- O consumo de energia da rede é reduzido pelo fato de que apenas a base emite requisições quando houver um evento. Além disso, a comunicação dos nós sensores acontece entre vizinhos, o que elimina a necessidade de endereçamentos e permite que cada nó faça a aglomeração de dados;
- Não há exigência de endereçamentos e topologias de rede globais. Além disso, as rotas são formadas apenas quando há necessidade, ou seja, quando há uma requisição, sendo assim, economiza-se energia da rede.

Os mesmos autores, também apontam as desvantagens encontradas ao se utilizar esse protocolo:

- Como o protocolo geralmente é baseado em uma topologia plana, existem problemas de escalabilidade e congestionamento (principalmente os nós localizados perto da base);
- Há um problema de *overhead* nos nós sensores durante os processos de correspondência de dados e requisições;
- Devido ao *flooding* necessário para propagar a requisição em cada nó, esse protocolo não é otimizado para eficiência energética e ainda necessita de

grandes quantidades de memória para armazenar o gradiente de requisições e mensagens recebidas;

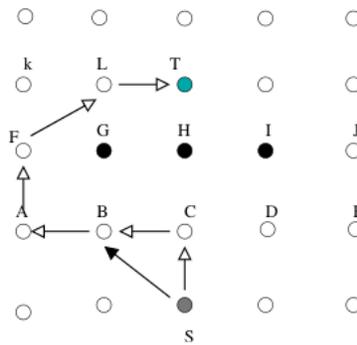
- A rota de caminho mais curto entre a base e o destinatário é selecionada mais vezes, o que acarreta em um maior desperdício de energia aos nós sensores pertencentes a essa rota e que por fim inutilizaria esses nós muito cedo.

### 2.3.3 Protocolo GEAR

O protocolo *Geographic and Energy Aware Routing* (GEAR) proposto por Yu, Govindan e Estrin (2001), é um exemplo de protocolo de roteamento geográfico e foi desenvolvido com técnicas para redução do consumo de energia em RSSF. O protocolo utiliza heurísticas e informações geográficas dos nós vizinhos para determinar a rota de comunicação entre o nó fonte e o destino final. Os autores ressaltam que cada nó sensor conhece sua própria localização e nível de energia restante, porém, a distância e nível de energia dos nós vizinhos é obtida aplicando-se um simples protocolo de Hello vizinho.

A localização de cada nó sensor também pode ser adquirida por baixo custo através da introdução de um GPS ou qualquer outro sistema de localização. Assim, através da informação geográfica e do nível de energia restante é realizado um cálculo de custo de envio, pois, esse custo será o parâmetro que definirá qual o melhor caminho a ser escolhido para se realizar a transmissão, isto é, o nó com menor custo de envio será escolhido. A cada nova transmissão, a função custo é recalculada a fim de se atualizar as informações de custo de cada nó sensor. Isto é feito porque a cada transmissão o nível de energia é reduzido. Além disso, a atualização permite uma otimização da rota escolhida e também possibilita aprender rotas em torno de “buracos“ (Figura 5), isto é, regiões onde os nós sensores

estão inutilizáveis; ou ”quando o nó sensor não possui nenhum outro vizinho mais próximo da região alvo além dele mesmo“ (AL-KARAKI; KAMAL, 2004).



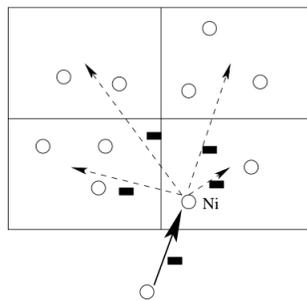
**Figura 5:** Aprender rotas em torno de ”buracos” (YU; GOVINDAN; ESTRIN, 2001).

No exemplo da Figura 5, os nós G, H e I são nós inoperantes e que portanto, são incapazes de transmitirem alguma informação. Nesse caso, quando C deseja transmitir alguma informação para o nó T, há a necessidade de atualizar as rotas. De forma que um novo caminho seja escolhido com a finalidade de contornar o problema gerado pela inoperação dos nós G, H e I.

Al-Karaki e Kamal (2004) resumem o algoritmo GEAR em duas etapas:

1. Encaminhar o pacote para a região alvo: uma vez recebido o pacote, o nó verificará se há um vizinho que está mais próximo da região alvo além dele mesmo. Se existe mais de um nó vizinho, o mais próximo à região alvo será escolhido para o próximo salto. Porém, se todos nós vizinhos estiverem mais distantes (há um ”buraco“ nessa região), o algoritmo selecionará um dos nós sensores para encaminhar o pacote baseando-se em seu custo de envio;

2. Encaminhar o pacote dentro da região: se um pacote alcançar a região, ele poderá ser difundido utilizando-se encaminhamento geográfico recursivo (*recursive geographic forwarding*), demonstrado pela Figura 6, ou poderá utilizar a técnica de *flooding* restrito (*restricted flooding*). Segundo os autores, o método de *flooding* restrito é melhor para redes que não são densas, enquanto que o primeiro é mais eficiente para redes com grande densidade.



**Figura 6:** Difusão de dados em uma região utilizando encaminhamento geográfico recursivo (YU; GOVINDAN; ESTRIN, 2001).

#### 2.3.4 Protocolo PROC

O protocolo PROC (*Proactive Routing with Coordination* proposto por Macedo *et al.* (2006) é um protocolo de roteamento proativo desenvolvido para redes de disseminação contínua.

Nesse protocolo forma-se uma árvore de roteamento chamada *backbone* que é uma rota unidirecional dos nós sensores até a base (ponto de acesso), conforme ilustrado na Figura 7c. Um *backbone* é composto por nós chamados de coordenadores que são responsáveis por encaminhar as mensagens até a base. Os outros nós, chamados de nós-folhas, comunicam-se diretamente com algum nó pertencente ao *backbone*. Assim, cada nó-folha precisa conhecer apenas o nó a qual se

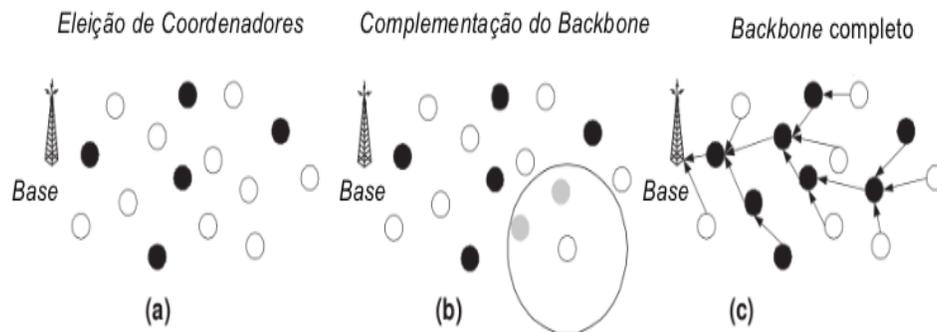
associa. Além disso, forma-se um novo *backbone* em intervalos regulares com o intuito de melhorar a economia de energia.

A formação do *backbone* no protocolo PROC se divide em duas fases: *eleição de coordenadores* e *complementação do backbone*.

Na primeira fase (*eleição de coordenadores*), os nós se auto-elegem coordenadores (Figura 7a). Cada nó tem uma probabilidade de se tornar coordenador que pode ser calculada por funções chamadas regras. Ao final da primeira fase cada nó sensor envia uma mensagem *Msync* para a rede, com o objetivo de informar seu novo estado. Quando um nó recebe uma mensagem *Msync* é verificada e estabelecida uma rota para alcançar a base. Assim, se um nó coordenador recebe uma mensagem deste tipo, ele irá selecionar seu nó pai entre os nós vizinhos coordenadores com a menor distância até a base. Caso não houver nenhum coordenador, ele selecionará um nó folha que satisfaça a condição de menor distância até a base. Este nó escolhido, então, se tornará coordenador. Quando houver dois ou mais nós que possuam a mesma distância até a base, o nó com maior energia residual é selecionado como nó pai.

Porém, essa fase pode não formar um *backbone* completo. Assim, a fase de *complementação do backbone* (Figura 7b) tem como princípio garantir que todos nós estejam conectados ao *backbone*, isto é, garante que exista uma rota de cada nó até a base. Nessa fase, então, há a inclusão de novos nós sensores ao *backbone* de maneira a garantir a completude do *backbone*. Para estabelecer uma rota até a base, cada nó folha verifica se existe um nó coordenador em sua vizinhança e será escolhido aquele que possuir a menor distância até a base. Caso não exista nenhum coordenador na vizinhança, o nó seleciona o seu vizinho com a menor

distância até a base, e envia uma mensagem chamada *MCoord* para forçá-lo a se tornar um nó coordenador.



**Figura 7:** Processo de criação do *backbone* no protocolo PROC. Fonte: adaptado de Macedo *et al.* (2006)

A Figura 7 exemplifica o processo de formação de um *backbone* utilizando-se o protocolo PROC. Os coordenadores são representados pelos nós pretos e os nós folha são representados pelos nós brancos. A primeira fase (*eleição de coordenadores*) seleciona um conjunto de nós comuns para se tornarem nós coordenadores (Figura 7a). Em seguida, procede-se, executando a segunda fase (*complementação do backbone*). A Figura 7b mostra um nó folha (nó branco) e sua área de alcance do rádio. Como não existe nenhum nó coordenador na sua vizinhança, o nó deve selecionar um dos seus vizinhos (nós cinzas) para se tornar um coordenador. Em seguida, as rotas são estabelecidas, como mostrada na Figura 7c.

### 3 METODOLOGIA

O objetivo geral deste trabalho foi desenvolver um protocolo de múltiplas camadas (*cross-layer*) para Redes de Sensores sem Fio(RSSF), envolvendo roteamento e acesso ao meio de comunicação. O intuito deste protocolo foi alcançar baixa latência e alta eficiência energética. Portanto, o objetivo geral pode ser classificado como método quantitativo, pois, alcançar estes objetivos equivale a comparar os valores obtidos nesse trabalho com valores encontrados no meio científico e assim concluir se os resultados são realmente melhores. O estudo bibliográfico de trabalhos relacionados foi importante para identificar métricas de avaliação dos protocolos de roteamento e enlace. Por exemplo, métricas que avaliarão a confiança na entrega de pacotes, eficiência energética, latência, o rendimento dos protocolos ou a influência dos ciclos de trabalho para a rede.

A linguagem de programação a ser utilizada no desenvolvimento do algoritmo *cross-layer*, foi a linguagem orientada a objetos, JAVA. Esta linguagem é amplamente aplicada nos problemas atuais por suas vantagens, como exemplos, pode-se citar a robustez e portabilidade, sendo que esta última vantagem permite que um programa desenvolvido em tal linguagem possa ser executado em qualquer plataforma que possua a máquina virtual JAVA.

A implementação do algoritmo proposto foi realizada utilizando-se o simulador Grubix, que é um simulador de Redes de Sensores sem Fio baseado em eventos. O simulador Grubix é baseado nas tecnologias Java e XML e adota os conceitos das camadas ISO/OSI. Ambos o simulador e a implementação do algoritmo *cross-layer* adotam a utilização da linguagem de programação JAVA principalmente pelas vantagens que ela possui.

O protocolo proposto envolve duas camadas (Rede e Enlace), portanto, foi necessário desenvolver um algoritmo de controle de acesso ao meio de comunicação (MAC) e mecanismos para roteamento. Atualmente, os protocolos baseados em contenção implementam ciclos de trabalho como meio de obter melhor eficiência energética, pois, através dessa técnica os nós sensores podem desligar seus rádios e economizar energia nos períodos de ociosidade. Neste projeto, também foi utilizada a técnica de ciclos de trabalho.

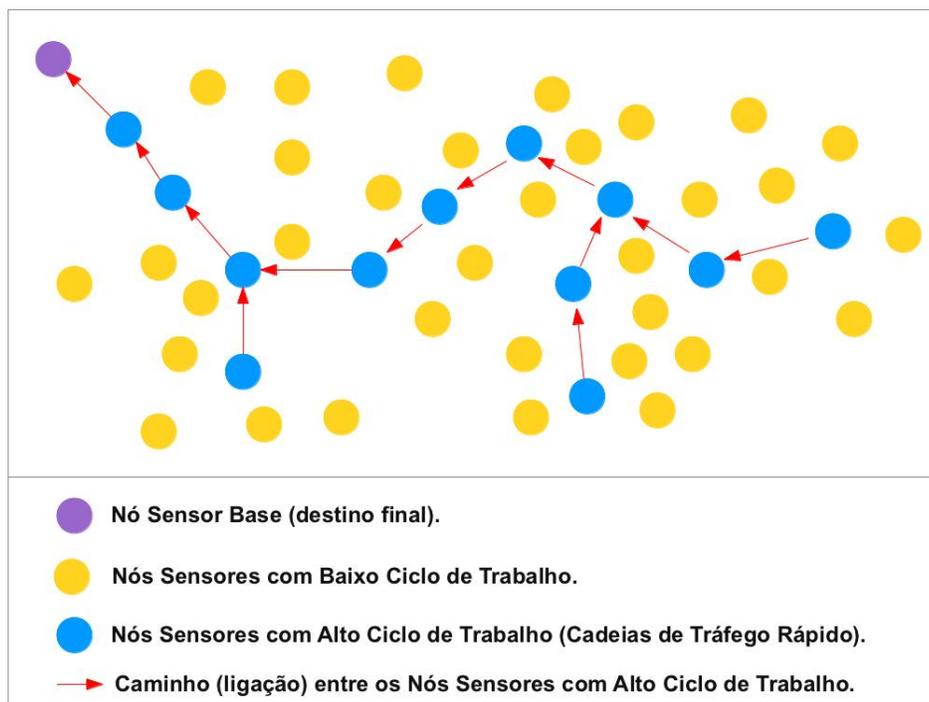
Apesar de economizarem energia, o uso de ciclos de trabalho atrasa a transmissão de mensagens na rede. Isto ocorre porque um nó que deseja transmitir uma mensagem deverá esperar que o nó receptor desperte e ligue seus rádios. Logo, a transmissão somente ocorrerá após essa espera. Portanto, atrasa-se a transmissão de mensagens, isto é, aumenta-se a latência da rede.

A proposta desse trabalho como mecanismo para reduzir a latência é a implantação de cadeias de tráfego rápido por onde as mensagens são transmitidas sem problemas decorrentes de latência. Em determinado momento, alguns nós são escalados para funcionarem em um modo mais ativo, ou seja, esses nós permanecerão ou com seus rádios o tempo todo ligados ou terão ciclos de trabalho alto. O grupo de nós sensores escolhidos formarão o *backbone* por onde as transmissões de mensagens ocorrerão de maneira ágil com objetivo de alcançar o nó base. Os demais nós sensores estarão em ciclo de trabalho reduzido com a finalidade de economizar energia. Em resumo, quando um nó desejar transmitir uma mensagem até a base bastará que este transmita a mensagem até o *backbone* por onde a mensagem será transmitida rapidamente.

Como os nós pertencentes à cadeia de tráfego rápido gastam mais energia que os demais, um mecanismo para balancear latência e eficiência em energia foi pro-

posto. Esse mecanismo permite que um novo *backbone* seja formado após certo tempo. Essa nova formação será constituída de outros nós sensores, possibilitando assim que a economia de energia seja fornecida a todos os nós da rede. Portanto, baixa latência e eficiência energética são os objetivos primários do protocolo proposto.

A figura 8 ilustra um exemplo de *backbone* composto por 12 nós sensores. Quando algum nó comum (nós de coloração amarela) deseja transmitir alguma mensagem, basta que este transmita a mensagem para o *backbone*. A partir deste, a mensagem fluirá livre de latência até a base.



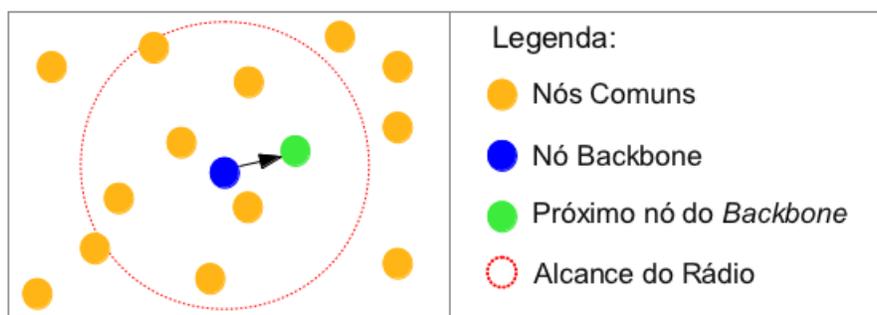
**Figura 8:** RSSF utilizando zonas de tráfego rápido.

O algoritmo de controle de acesso ao meio utilizado para controlar os ciclos de trabalho é o protocolo X-MAC. Este algoritmo foi implementado no simulador Grubix. Já o controle da formação da rede bem como o controle dos roteamentos, é exercido pelo algoritmo *cross-layer* proposto neste trabalho.

Foram propostas otimizações na formação do *backbone*, proporcionando novas versões e variações do algoritmo *cross-layer*. As otimizações propostas são especificadas a seguir.

O número de nós escalados para pertencer ao *backbone* é definido previamente. E essa escolha permite obter uma rede que economize mais energia e reduza a latência da rede de maneira balanceada.

No algoritmo básico forma-se primeiramente o *backbone* e em seguida associa-se cada nó comum a um nó *backbone* ou a outro nó comum. Tais associações formarão na rede diversos gradientes, de acordo com o alcance e número de saltos necessários para transmitir um pacote até o *backbone*. Nesse primeiro algoritmo, a escolha dos nós sensores para formar o *backbone* é feita de forma aleatória (Figura 9).



**Figura 9:** Exemplo de inclusão aleatória de um nó comum ao *backbone*.

No exemplo da Figura 9 ilustra-se o momento em que um nó *backbone* (nó azul) deverá escolher o próximo nó a ser incluído ao *backbone*. Seu rádio possui um alcance que é ilustrado pelo círculo vermelho. No algoritmo de inclusão aleatória permite-se que seja escolhido qualquer um dos nós dentro do alcance do rádio. Nesse caso (Figura 9), o nó de cor azul escolheu aleatoriamente o nó de coloração verde como o próximo nó do *backbone*.

A primeira otimização desse algoritmo realizará a formação da rede em forma de árvore, isto é, a formação do *backbone* envolve a criação de ramos com o objetivo de encontrar caminhos mais curtos até a base. Tais caminhos, quando encontrados, reduzirão a latência da rede.

A segunda otimização permitirá a criação de novas rotas a partir do acréscimo de novos nós ao *backbone*. Porém, esse acréscimo somente é aceito quando o nó a ser adicionado unir dois nós *backbone* distantes o bastante para satisfazer o valor mínimo definido previamente nas configurações do algoritmo. Aliado à otimização anterior esse mecanismo contribuirá para a rapidez da entrega de pacotes e consequentemente reduzirá ainda mais a latência da rede.

A terceira otimização permitirá que os nós sensores escalados ao *backbone* possam se tornar nós comuns novamente quando não forem considerados importantes para o *backbone*. Um nó é considerado importante quando algum outro nó utiliza-o como rota para se alcançar a base. Aqueles que não forem importantes poderão se tornar comuns novamente, definindo como baixo o seu ciclo de trabalho.

As Figuras 10 e 11 ilustram um *backbone* formado sem utilizar as três otimizações citadas anteriormente e um *backbone* formado utilizando essas três otimizações, respectivamente.



**Figura 10:** Exemplo de um *backbone* formado sem otimizações.

Na Figura 10, o caminho percorrido pelas mensagens transmitidas ao *backbone* até a base (nó vermelho) segue um trajeto único, indicado pelas setas vermelhas. Nesse caso, demonstra-se uma rede sem ramificação e sem adição ou remoção de nós do *backbone*. Note, que há um nó comum de cor verde que é alcançado pelo rádio de dois nós *backbone*. Se incluído ao *backbone* este nó criará um caminho mais curto até a base. Existem também dois nós de coloração ciano pertencentes ao *backbone* que poderão voltar a ser nós comuns novamente porque não mais serão utilizados como caminho até a base.

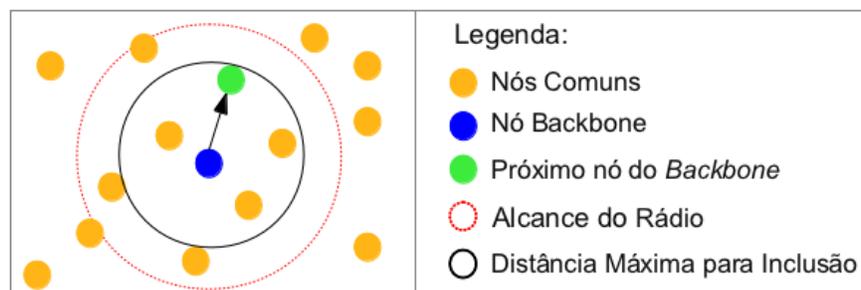


**Figura 11:** Exemplo de um *backbone* formado com otimizações.

Já na Figura 11, formou-se uma ramificação no *backbone* a partir da adição de um novo nó ao *backbone* (nó verde). Essa inclusão permitiu que um caminho

mais curto até a base fosse formado. Além disso, os dois nós de coloração ciano que antes pertenciam ao *backbone* na Figura 10, agora, foram removidos do *backbone* deste exemplo (Figura 11). Note que não há mais setas vermelhas (indicação das direções das transmissões) nestes nós. Isto, porque os demais nós na região conseguem se comunicar com outros nós do *backbone* que estão mais próximos à base. Essa proximidade corresponde ao número de saltos necessários para que uma mensagem partindo de um certo nó *backbone* alcance a base.

Na quarta otimização, foi alterado a forma como ocorre a formação da rede. No algoritmo básico emprega-se a aleatoriedade na escolha dos nós sensores que irão compor o *backbone*, o que muitas vezes gera resultados ruins. Será considerado ruim o *backbone* que não cobrir boa parte da rede, isto é, se muitos nós comuns necessitarem de dois ou mais saltos para se alcançar o *backbone*. Nesta otimização, entretanto, aplicar-se-á noções de distância e RSSI como forma de escolher os nós que terão ciclo de trabalho alto. Definiu-se uma distância máxima para a inclusão de um nó sensor ao Backbone, isto é, somente poderão ser escolhidos os nós que estejam dentro do raio de alcance delimitado pela distância máxima. Através dessa técnica há uma melhor redução da latência da rede. A Figura 12, a seguir, ilustra uma inclusão de nós ao *backbone* limitada por uma distância máxima.



**Figura 12:** Exemplo de inclusão de um nó comum ao *backbone* de acordo com a Distância Máxima.

Nesse exemplo (Figura 12), existem dois raios de alcance: um é o alcance total do rádio do nó sensor e o outro é um alcance menor que limita quais nós podem ser incluídos ao *backbone*. Esse segundo alcance, entretanto, também poderia ser do tamanho do alcance total do rádio. Mas, nesse exemplo, optou-se por um raio de alcance menor para melhor demonstrar a quarta otimização. Note, que o número de opções de escolha para o nó *backbone* (nó de cor azul) foi diminuído. Este nó, portanto, deverá escolher o nó mais distante dentre suas opções e que, neste exemplo, será o nó comum de coloração verde.

Finalizada a etapa de desenvolvimento, diversas simulações foram desenvolvidas com o objetivo de comparar e avaliar o algoritmo e suas otimizações. As simulações foram desenvolvidas utilizando o simulador Grubix e envolverão grande número de nós sensores (mais de 100). Neste caso, utilizou-se um total de 120 nós sensores. Sendo que um nó (base local) é sempre fixado em uma mesma posição e os demais são distribuídos aleatoriamente por uma área de 200m x 185m a cada nova simulação. Cada nó sensor, possui um rádio com alcance de 40m.

A cada simulação, observou-se a formação da rede final e qual era a influência da variação dos valores de algumas variáveis de configuração para a rede. Logo, diversos valores foram utilizados e computados de forma a obter os resultados finais. Os valores das seguintes variáveis sofrerão alterações durante as simulações:

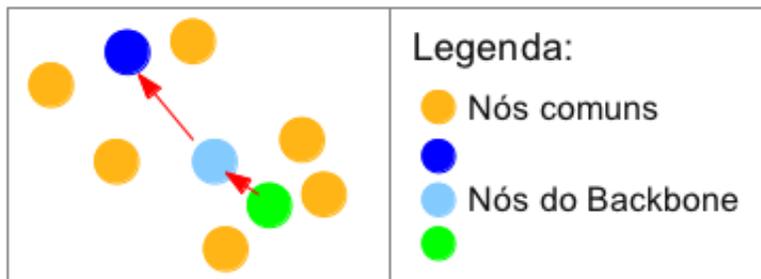
- Número de nós sensores escalados para pertencer ao *backbone*: 10, 15, 20, 25 e 30 nós iniciais;
- Distância máxima para inclusão de um nó sensor ao *backbone*: 60%, 70%, 80%, 90% e 100% do alcance total dos rádios dos nós sensores;

Para cada um desses valores foram feitas 100 simulações com a finalidade de coletar os dados necessários para conclusão deste trabalho.

O número de nós escalados para pertencer ao *backbone* é responsável pela redução da latência da rede porque permite que mais nós comuns possuam acesso ao *backbone* com transmissões de apenas um salto. Assim, criar um *backbone* com menos de 10 nós não fornece uma boa cobertura, pois, muitos nós comuns precisarão realizar mais de uma transmissão para alcançar o *backbone*, o que não será vantajoso para redução da latência. Já valores acima de 30 nós, geram uma saturação de nós *textitbackbones* na rede, isto é, muitos nós incluídos ao *backbone* nem sequer serão utilizados como caminho até a base. Por isso, utilizou-se os valores citados anteriormente (10, 15, 20, 25 e 30 nós iniciais).

Na formação em que utiliza-se inclusão aleatória, há casos em que adiciona-se muitos nós desnecessários e que não serão úteis aos nós comuns na região. O exemplo da figura 13 ilustra um caso em que a inclusão aleatória adicionou um nó ao *backbone* desnecessariamente. Nesse exemplo, o nó de coloração verde foi adicionado ao *backbone* e encontra-se perto de um outro nó (cor azul claro), que também já foi incluído ao *backbone*. Os nós comuns naquela região possuem um rádio que permite uma comunicação com ambos os nós citados. Assim, não há necessidade de dois nós *backbone* nesta região.

Já a formação em que inclui-se um nó sensor ao *backbone* de acordo com a distância máxima permitida tem como princípio evitar que este tipo de situação ocorra. Nesse caso, o próximo nó a ser adicionado ao *backbone* deve ser aquele mais distante possível dentro do limite de inclusão. O objetivo é fornecer uma melhor cobertura do *backbone* aos nós comuns. Valores de distância máxima inferiores a 60% do alcance total do rádio, funcionam como uma inclusão aleatória em



**Figura 13:** Exemplo de inclusão desnecessária de nós sensores ao *backbone*.

que os nós adicionados ao *backbone* estão muito próximos uns dos outros. Dessa forma, a cobertura do *backbone* não será tão eficiente. Portanto, utiliza-se neste trabalho valores superiores a 60% do alcance total do rádio.

### 3.1 Formação do *Backbone*

A formação do *backbone* inicia-se a partir do nó base. Esse nó, escolherá dentre todos os seus vizinhos o primeiro nó sensor a ser incluído ao *backbone*. A forma como escolhe-se o próximo nó depende do método utilizado, sendo eles: inclusão aleatória ou inclusão de acordo com a distância máxima permitida.

Após a escolha, transmite-se ao nó escolhido, uma mensagem chamada “*Token*”. Nessa mensagem requisita-se que o nó se torne nó *backbone*. Os demais nós que escutam o “*Token*” serão impedidos de se tornarem nós *backbone*. Porém, uma exceção ocorre quando todos os nós em uma certa região já receberam este tipo de mensagem e nesse caso, escolhe-se o nó que tenha recebido menos vezes essa mensagem. Os nós incluídos ao *backbone* receberão um identificador que indicará a posição deste nó no *backbone*, sendo que o valor atribuído corresponderá ao nú-

mero de saltos necessários para que uma mensagem partindo deste nó alcance a base.

A formação do *backbone* prossegue com os nós sensores que são, um a um, incluídos ao *backbone*. Cada um destes, também escolherá o próximo nó sensor a ser adicionado ao *backbone* e transmite ao nó escolhido, uma mensagem “*Token*”. O processo de inclusão de nós sensores encerra-se quando o número inicial de nós escalados para formar o *backbone* é atingido.

Na sequencia, o último nó incluído, deverá alertar a base que o número de nós escalados para pertencer ao *backbone* já foi atingido. Para isso, o último nó encaminhará uma mensagem que percorrerá cada nó incluído ao *backbone*. Durante esta etapa, permite-se a ramificação e a inclusão de novos nós sensores ao *backbone* com o intuito de encurtar o caminho até a base.

Quando a mensagem chega a base, inicia-se a última etapa da formação do *backbone*. Nessa etapa, cada nó pertencente ao *backbone* emitirá uma mensagem em *broadcasting* para otimizações e que fornecerá uma melhoraria nas rotas até a base, isto é, ainda há a possibilidade de ramificar mais a rede. Durante as transmissões dessas mensagens, os nós comuns que estão a um salto do *backbone* escutarão as mensagem transmitidas nessa etapa. Cada um destes nós se associará ao nó *backbone* mais próximo à base (nó *backbone* com menor identificador).

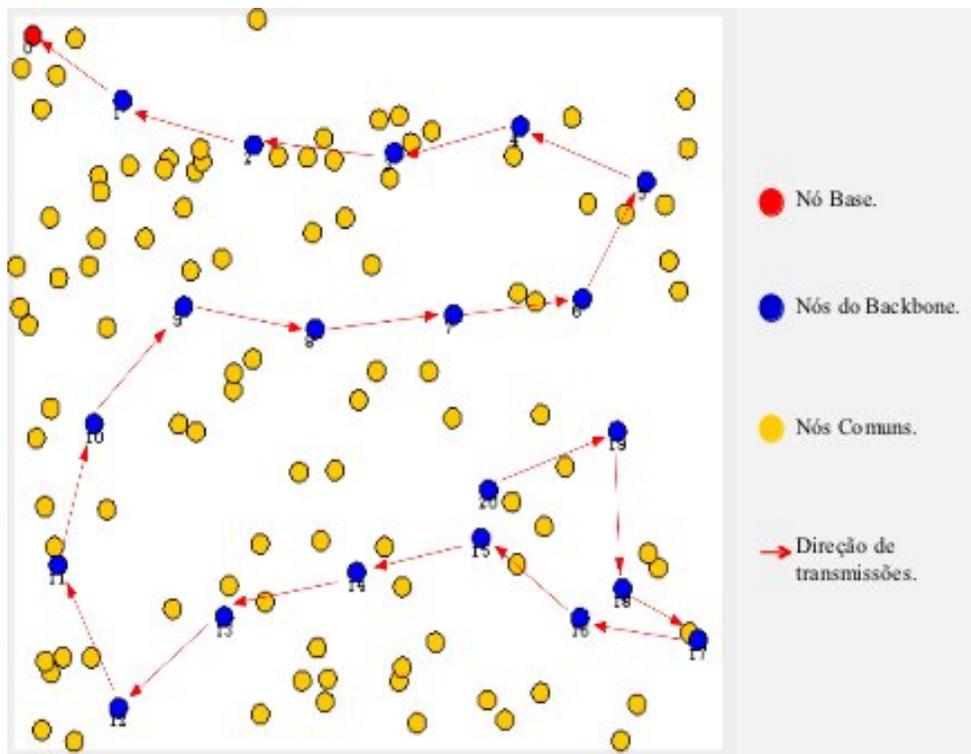
Em seguida, cada nó comum a um salto do *backbone* emitirá uma mensagem em *broadcasting*. O objetivo será informar aos nós distantes a dois saltos do *backbone* sobre o término na formação do *backbone*. Assim, os nós distantes a dois saltos do *backbone* poderão se associar a um nó comum que alcance o *backbone* com apenas um salto. Agora, cada nó distante a dois saltos precederá da mesma forma, e alertarão os nós distantes a três saltos. Estes últimos, por sua vez, aler-

tarão os distantes a quatro saltos e assim por diante. Isto prossegue-se até que todos os nós comuns distantes a mais de um salto estejam conectados e consigam transmitir mensagens para a base.

Ao final dessas etapas a rede já estará formada e pronta para iniciar o monitoramento do ambiente.

## 4 RESULTADOS

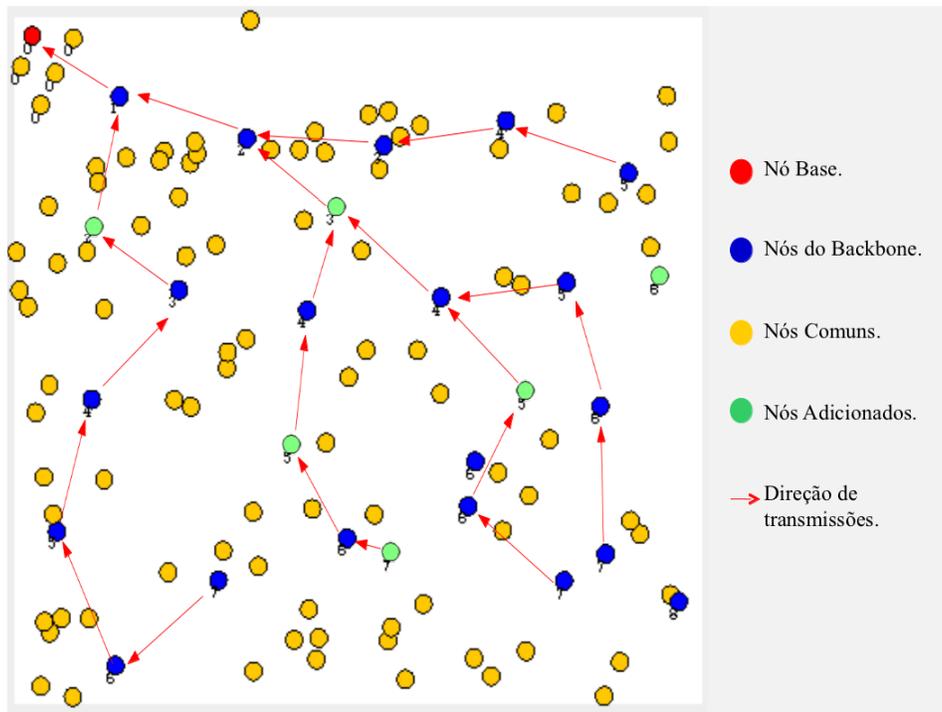
O protocolo proposto e as otimizações definidas na metodologia foram implementados no simulador Grubix. As Figuras 14 e 15 a seguir ilustram um *backbone* construído com o protocolo desenvolvido sem utilizar as otimizações propostas e utilizando as otimizações propostas, respectivamente. Nesse exemplo, optou-se por criar um *backbone* de tamanho inicial 20.



**Figura 14:** Exemplo de uma rede e *Backbone* formados sem otimizações.

O *backbone* ilustrado pela Figura 14 é constituído de três tipos de nós: comuns (amarelos), nós pertencentes ao *backbone* (azuis) e o nó base (vermelho). As ligações (comunicação) entre os nós sensores pertencentes a esse *Backbone* e as

direções das transmissões das mensagens são indicadas pelas setas vermelhas. Neste primeiro exemplo (Figura 14), não utilizou-se nenhuma das otimizações, portanto, o caminho para as transmissões de mensagens segue uma única via e direção.



**Figura 15:** Exemplo de uma rede e *Backbone* formados com otimizações.

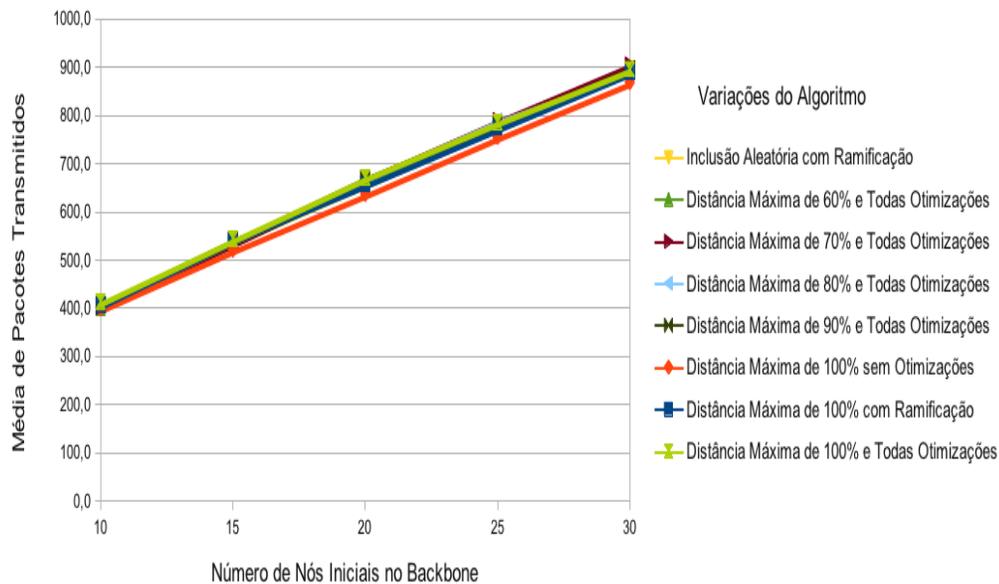
Já o *backbone* ilustrado pela Figura 15 é constituído de quatro tipos de nós: comuns (amarelos), nós pertencentes ao *backbone* (azuis), nós adicionados ao *backbone* com o intuito de criar atalhos (verde) e o nó base (vermelho). As ligações (comunicação) entre os nós sensores pertencentes ao *Backbone* e as direções das transmissões das mensagens são indicadas pelas setas vermelhas. Na formação do *backbone* deste exemplo (Figura 15), utilizou-se ramificação e adição/remoção de nós, assim, pode-se ver diversos ramos e caminhos até a base. Um indicativo de que houve ramificação em diversos pontos é a existência de nós com identifi-

res *backbone* iguais, por exemplo, existem vários nós com identificador 4. Note, ainda, que há três nós que estão indicados como pertencentes ao *Backbone*, mas que não possuem setas saindo ou chegando a eles. Isto significa que nenhum nó utiliza-os como caminho de transmissão de mensagens. Portanto, estes nós poderão se tornar nós comuns novamente para que haja uma melhor economia de energia.

As Tabelass (1, 2, 3, 4, 5), a seguir, representam a média e o desvio padrão para cada um dos testes realizados. Nessas tabelas, os “Métodos” especificados equivalem à variação do algoritmo proposto. Sendo:

- Método 1 - Distância Máxima de 100% com Ramificação
- Método 2 - Distância Máxima de 100% sem Otimizações
- Método 3 - Inclusão Aleatória com Ramificação
- Método 4 - Distância Máxima de 60% e Todas Otimizações
- Método 5 - Distância Máxima de 70% e Todas Otimizações
- Método 6 - Distância Máxima de 80% e Todas Otimizações
- Método 7 - Distância Máxima de 90% e Todas Otimizações
- Método 8 - Distância Máxima de 100% e Todas Otimizações

Foi verificada a relação entre número de pacotes transmitidos para formar o *backbone* e o número inicial de nós previstos para compor o *backbone*. Essa relação é demonstrada na Figura 16 e os valores obtidos para essa métrica foram especificados na Tabela 1.



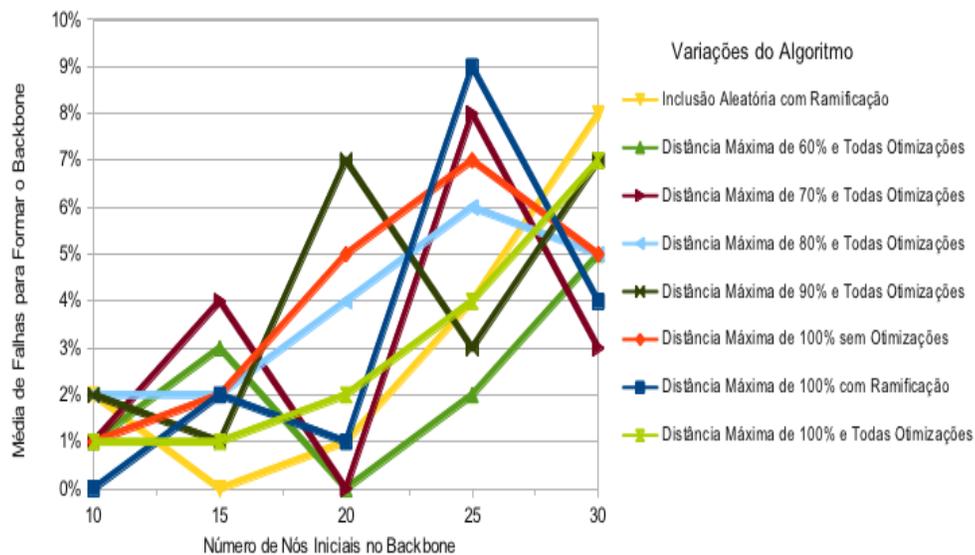
**Figura 16:** Média de pacotes transmitidos para formação do *backbone*

**Tabela 1:** Média e Desvio Padrão (DP) para número de pacotes transmitidos para formação do *backbone*.

	Número de nós iniciais no <i>backbone</i>									
	10		15		20		25		30	
	Média	DP	Média	DP	Média	DP	Média	DP	Média	DP
Método 1	408,15	30,55	543,16	33,67	658,25	41,11	774,38	45,11	890,53	59,61
Método 2	399,59	38,37	521,73	39,90	637,22	42,58	755,51	37,65	869,39	50,43
Método 3	408,34	38,37	539,59	50,63	658,63	44,45	776,63	57,22	897,83	39,98
Método 4	400,46	43,93	537,20	55,68	663,87	56,73	783,16	51,60	896,07	51,09
Método 5	402,32	46,75	536,24	46,75	669,53	51,14	790,84	54,31	906,42	48,20
Método 6	407,98	42,06	541,47	43,16	667,82	50,06	789,72	49,35	890,32	49,38
Método 7	410,04	40,81	540,63	43,95	671,76	46,27	783,44	46,66	898,96	58,87
Método 8	413,49	35,25	543,86	41,05	671,89	35,31	786,90	44,87	896,61	51,63

Conforme demonstra a Figura 16, o aumento do número de pacotes transmitidos para formar a rede é linear. Por exemplo, para 25 nós iniciais, obteve-se uma média de 800 pacotes transmitidos. O número médio de pacotes transmitidos du-

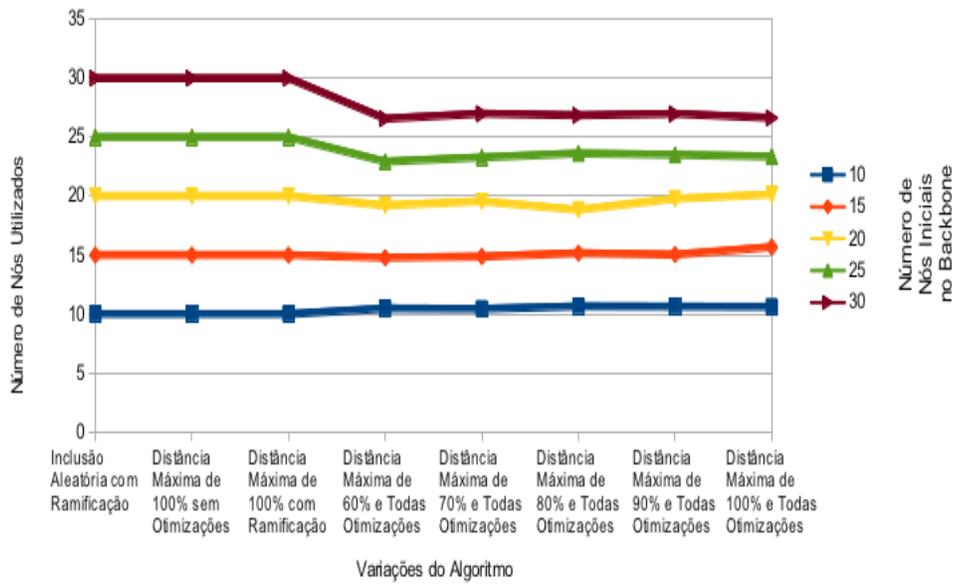
rante a formação da rede é importante, pois, permite avaliar o custo de formação da rede e quanto de energia se gasta nesta formação.



**Figura 17:** Média de falhas para formar o *backbone*

O gráfico da Figura 17 ilustra a média de falhas encontradas durante a formação da rede. Estas falhas aconteceram devido ao problema de borda, isto é, durante a formação do *backbone* haviam locais onde todos os nós já pertenciam ao *backbone*. Portanto, não houve mais formas de continuar a formação, pois, o número estipulado no início devia ser obrigatoriamente cumprido. Acima de 30 nós sensores houve uma dificuldade na criação das redes justamente por este problema, mostrando que havia uma saturação de nós *backbones*. Uma forma de resolver esse problema é permitir que a formação do *backbone* seja suspensa quando ocorrer a interrupção, passando para a próxima etapa da formação da rede. Em geral, o aumento do número de nós iniciais do *backbone* tende a aumentar a ocorrência de tais erros porque aumenta a chance de ocorrer problemas de borda. Essas fa-

lhas contribuem com o aumento do gasto de energia da rede, entretanto, em nossos testes em média menos de 10% das simulações de formação da rede falharam.



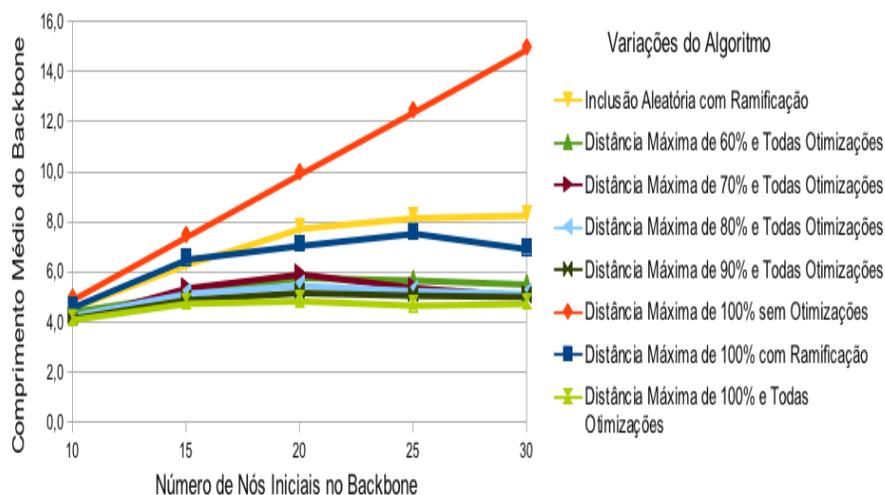
**Figura 18:** Número médio de nós necessários para formar o *backbone*

**Tabela 2:** Média e Desvio Padrão (DP) para número de nós necessários para formar o *backbone*

	Número de nós iniciais no <i>backbone</i>									
	10		15		20		25		30	
	Média	DP	Média	DP	Média	DP	Média	DP	Média	DP
Método 1	10,00	0,00	15,00	0,00	20,00	0,00	25,00	0,00	30,00	0,00
Método 2	10,00	0,00	15,00	0,00	20,00	0,00	25,00	0,00	30,00	0,00
Método 3	10,00	0,00	15,00	0,00	20,00	0,00	25,00	0,00	30,00	0,00
Método 4	10,47	0,84	14,78	1,56	19,21	1,87	22,93	2,29	26,57	2,96
Método 5	10,44	1,08	14,88	1,82	19,55	1,99	23,28	2,67	26,99	2,50
Método 6	10,65	1,05	15,16	1,50	18,83	1,95	23,62	2,63	26,85	2,83
Método 7	10,63	1,06	15,04	1,83	19,77	2,18	23,51	2,49	26,98	2,71
Método 8	10,62	1,03	15,68	1,65	20,13	2,20	23,35	2,62	26,63	3,11

O número médio de nós utilizados na criação na criação de um *backbone* é ilustrado pela Figura 18 e os resultados obtidos utilizando essa métrica são es-

pecificados na Tabela 2. O valor final de nós sensores pertencentes ao *backbone* modifica-se somente nos algoritmos em que aplica-se a técnica de acréscimo e remoção de nós sensores da rede. Nos demais casos o valor final é o estipulado pelas configurações iniciais. Através dessa técnica foi possível constatar que o número ideal de nós sensores em uma rede para este cenário é em média de 26 nós. Aumentando-se mais do que esse valor satura-se a rede de nós com alto consumo de energia e aumenta-se as chances de ocorrerem erros de formação do *backbone*.



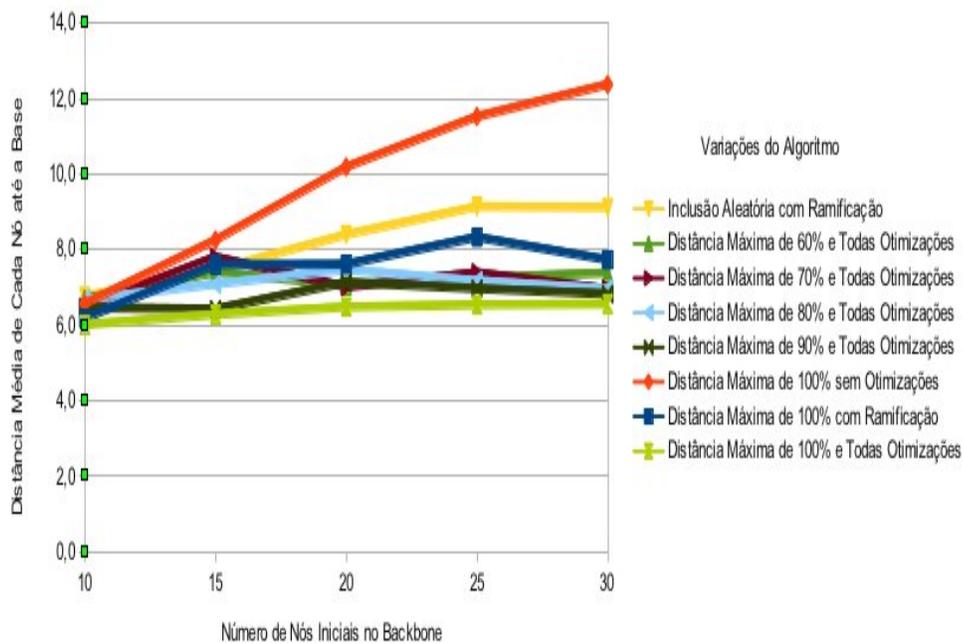
**Figura 19:** Comprimento médio do *backbone*

A adição e/ou remoção de nós aliado à ramificação dos caminhos até a base auxiliam na redução da latência da rede, pois, permite que uma mensagem transmitida seja encaminhada até a base mais rapidamente (Figura 19). Os valores obtidos para a métrica “Comprimento do Backbone” são especificados na Tabela 3. A partir dessa figura é possível notar uma grande redução no comprimento médio do *backbone* quando o algoritmo de ramificação é aplicado e reduz-se ainda mais quando aplica-se o método de adição/remoção de nós. Isto demonstra, assim, a eficiência desses mecanismos. Além disso, é possível verificar que o mecanismo para forma-

**Tabela 3:** Média e Desvio Padrão (DP) para comprimento do *backbone*.

	Número de nós iniciais no <i>backbone</i>									
	10		15		20		25		30	
	Média	DP	Média	DP	Média	DP	Média	DP	Média	DP
Método 1	4,68	0,62	6,58	1,09	7,13	1,97	7,61	2,24	6,99	2,11
Método 2	5,00	0,04	7,50	0,04	10,00	0,00	12,47	0,16	14,98	0,10
Método 3	4,65	0,57	6,38	1,19	7,81	1,84	8,24	2,85	8,34	2,48
Método 4	4,49	0,69	5,21	1,38	5,85	1,61	5,75	1,41	5,58	1,18
Método 5	4,18	0,92	5,44	1,27	5,99	1,64	5,45	1,38	5,12	0,78
Método 6	4,33	0,74	5,22	1,35	5,54	1,37	5,32	1,10	5,21	1,04
Método 7	4,23	0,85	4,96	1,29	5,25	1,34	5,14	1,15	5,09	0,95
Método 8	4,18	0,79	4,82	1,10	4,91	1,21	4,74	0,74	4,81	0,78

ção do *backbone* em que são utilizados limites de distância para inclusão de nós obtiveram os melhores resultados.

**Figura 20:** Distância média de cada nó até a base

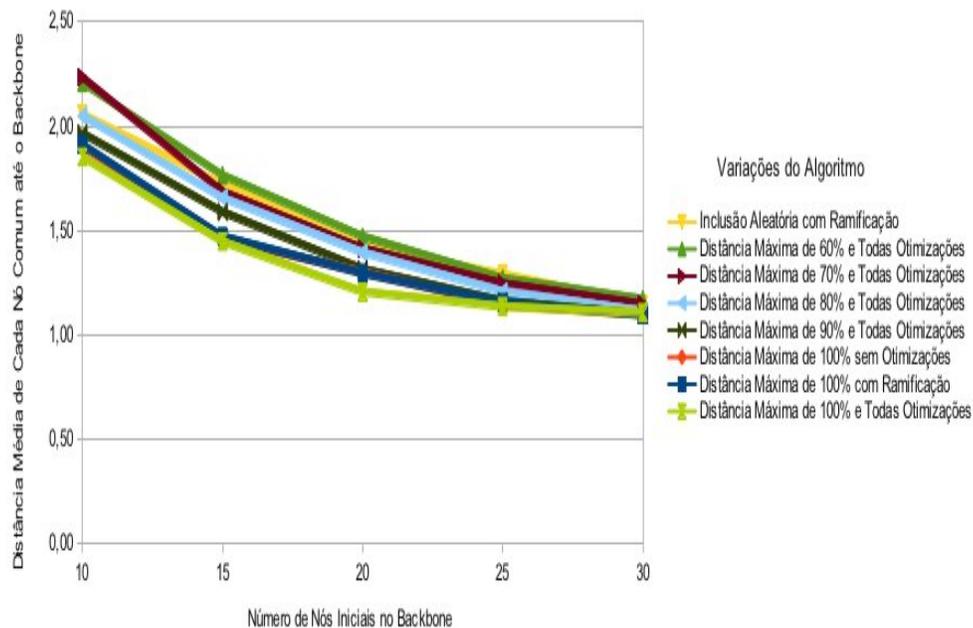
**Tabela 4:** Média e Desvio Padrão (DP) para distância de cada nó até a base.

	Número de nós iniciais no <i>backbone</i>									
	10		15		20		25		30	
	Média	DP	Média	DP	Média	DP	Média	DP	Média	DP
Método 1	6,19	0,91	7,61	1,02	7,62	1,53	8,34	1,50	7,74	1,52
Método 2	6,58	0,85	8,26	1,14	10,20	1,38	11,53	1,62	12,39	1,43
Método 3	6,77	0,72	7,47	1,27	8,41	1,58	9,14	2,09	9,13	1,92
Método 4	6,71	0,63	7,41	1,11	7,07	1,30	7,26	1,18	7,39	0,97
Método 5	6,49	0,78	7,81	1,17	7,02	1,43	7,40	1,13	6,96	0,61
Método 6	6,69	0,78	7,10	1,16	7,47	1,15	7,18	0,95	6,96	0,87
Método 7	6,47	0,75	6,43	0,98	7,14	1,15	6,98	1,01	6,83	0,80
Método 8	6,02	0,74	6,29	0,96	6,49	1,02	6,55	0,59	6,57	0,57

Os valores obtidos para a métrica “Distância de Cada Nó até a Base” são especificados na Tabela 4 e na Figura 20 é possível visualizar o efeito da adição/remoção de nós e da ramificação do *backbone* na diminuição da distância média de cada nó sensor até a base (*sink*). Por exemplo, quando utiliza-se 25 nós iniciais para formar o *backbone* e aplica-se o algoritmo de limite de inclusão igual a 100% e todas as otimizações propostas obtêm-se uma redução de aproximadamente 43.2% em relação ao algoritmo de limite de inclusão igual a 100% e nenhuma das otimizações. E uma redução de aproximadamente 28.5% em relação ao algoritmo de inclusão aleatória com ramificação. Além disso, aplicando-se tais mecanismos manteve-se a distância média em torno de 6 saltos a medida que aumentou-se o número de nós iniciais do *backbone*. Ressalta-se ainda que mesmo no pior caso (quando não há ramificação da rede), a média de saltos de cada nó até a base é inferior a 14 saltos.

Como a distância média de cada nó até a base se mantém constante para o melhor caso (algoritmo com limite de inclusão igual a 100% e todas as otimizações propostas), haveria de se dizer que tanto faz utilizar 10 ou 25 nós iniciais. Entre-

tanto, a justificativa para se aumentar o número de nós iniciais se deve à redução da latência proporcionada quando são escalados um número maior de nós sensores para pertencer ao *backbone*.



**Figura 21:** Distância média de cada nó comum até o *backbone*

**Tabela 5:** Média e Desvio Padrão (DP) para distância de cada nó comum até o *backbone*.

	Número de nós iniciais no <i>backbone</i>									
	10		15		20		25		30	
	Média	DP	Média	DP	Média	DP	Média	DP	Média	DP
Método 1	,91	0,37	1,47	0,27	1,29	0,24	1,15	0,15	1,09	0,10
Método 2	1,90	0,37	1,46	0,30	1,30	0,22	1,14	0,13	1,10	0,12
Método 3	2,06	0,36	1,72	0,33	1,43	0,26	1,29	0,24	1,15	0,13
Método 4	2,21	0,38	1,76	0,29	1,47	0,22	1,27	0,16	1,17	0,13
Método 5	2,23	0,37	1,68	0,29	1,41	0,24	1,24	0,16	1,15	0,13
Método 6	2,05	0,37	1,66	0,29	1,39	0,23	1,21	0,15	1,11	0,12
Método 7	1,97	0,34	1,59	0,35	1,31	0,21	1,16	0,15	1,10	0,11
Método 8	1,85	0,36	1,45	0,30	1,20	0,18	1,13	0,13	1,10	0,11

A distância média de cada nó até a base (Figura 20) é responsável pelo aumento ou a redução do consumo de energia. Já a distância média de saltos (Figura 21) para se alcançar o *backbone* é responsável pelo aumento ou redução da latência. Portanto, essas duas métricas são importantes para avaliar a eficiência do protocolo proposto. Os valores obtidos para a métrica “Distância de Cada Nó Comum até o *Backbone*“ são especificados na Tabela 5. E a Figura 21 ilustra o efeito da aplicação dos mecanismos de otimização propostos neste trabalho. A partir destes resultados obteve-se que a média de saltos de cada nó sensor até o *backbone* é em média 1,13, o que equivale a uma média 13% maior que a média de saltos do algoritmo ótimo (quando todos os nós sensores estão a um salto do *backbone*).

## 5 CONCLUSÃO

O uso de ciclos de trabalho em RSSF proporcionam uma redução no consumo de energia. Porém, como foi visto, esse mecanismo resulta em uma maior latência na rede. Assim, balancear latência e eficiência em energia é uma tarefa conflitante.

Para resolver o problema da latência, pode-se utilizar um algoritmo de formação de um *backbone*. Esse mecanismo, fornece uma rede quase totalmente livre de latência. Os nós sensores que compõem o *backbone* garantem uma entrega rápida de mensagens porque estão em alto ciclo de trabalho, o que significa redução de latência. Os demais nós sensores, que estão em baixo ciclo de trabalho para economizarem energia, utilizam o *backbone* como rota de transmissões de mensagens até a base.

O algoritmo proposto faz uso dessas duas técnicas (ciclos de trabalho e formação de um *backbone*), portanto, balanceou-se latência e eficiência em energia. Em nosso algoritmo, para uma rede de 25 nós sensores a distância de cada nó comum até o *backbone* foi em média de 1.13 saltos. Isto corresponde a 13% a mais de latência que o algoritmo ótimo (quando todos nós comuns alcançam o *backbone* com apenas um salto). Além disso, o algoritmo também possui baixo custo de formação e baixa taxa de falhas. De forma geral, o número de pacotes transmitidos durante a formação da rede cresce linearmente com a quantidade de nós previstos inicialmente para formar o *backbone* e ocorreram menos de 10% de falhas na formação do *backbone*.

## 5.1 Trabalhos Futuros

Uma RSSF necessita de um bom projeto em eficiência em energia e alguns caso uma latência reduzida. Este trabalho foi proposto com o intuito de balancear estes dois requisitos conflitantes. O protocolo proposto será utilizado posteriormente em pesquisas do Grupo de Pesquisa de Redes Ubíquas (Grubi) do Departamento de Ciência da Computação da Universidade Federal de Lavras (UFLA) em simulações em que os requisitos citados são importantes.

Pretende-se melhor algoritmo reduzindo a taxa de falhas na formação do *backbone* e reduzir o consumo de energia do *backbone* sem aumentar a latência.

## Referências

AL-KARAKI, J. N.; KAMAL, A. E. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*, v. 11, p. 6–28, 2004.

ALI, N.; ELSAYED, H.; EL-SOUDANI, M.; AMER, H. Single and multi coding schemes for efficient wireless sensor networks. *Wireless Sensor Network*, v. 4, p. 52–58, 2012.

BUETTNER, M.; YEE, G. V.; ANDERSON, E.; HAN, R. X-mac: A short preamble mac protocol for duty-cycled wireless sensor network. *SenSys'06*, v. 6, 2006.

EL-HOIYDI, A. Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks. *Proceedings of IEEE International Conference on Communications*, 2002.

HEIDEMANN, J.; GOVINDAN, R. Handbook of networked and embedded control systems. In: \_\_\_\_\_. [S.l.]: Springer Verlag, 2004. cap. Embedded Sensor Networks, p. 18p.

HEINZELMAN, W. R.; CHANDRAKASAN, A.; BALAKRISHNAN, H. Energy-efficient communication protocol for wireless microsensor networks. In: *Proceedings of the 33rd Hawaii International Conference on System Sciences - Volume 8*. [S.l.: s.n.], 2000.

HILL, J.; HORTON, M.; KLING, R.; KRISHNAMURTHY, L. The platforms enabling wireless sensor networks;. *Communications of the ACM*, v. 47, p. 41–46, 2004.

IEEE. *IEEE 802.11 – Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY)*. 1999.

IEEE. *IEEE 802.15.4 - Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*. 2006.

INTANAGONWIWAT, C.; GOVINDAN, R.; ESTRIN, D. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In: *MOBICOM*. [S.l.]: ACM, 2000. p. 56–67.

IYER, A.; KULKARNI, S.; MHATRE, V.; ROSENBERG, C. A taxonomy-based approach to design of large-scale sensor networks. In: LI, Y.; THAI, M.; WU, W. (Ed.). *Wireless Sensor Networks and Applications*. [S.l.]: Springer US, 2008, (Signals and Communication Technology). p. 3–33. ISBN 978-0-387-49591-0.

JANEK, J. F.; EVANS, J. J. Predicting ground effects of omnidirectional antennas in wireless sensor networks. *Wireless Sensor Network*, v. 2, p. 879–890, 2010.

LI, Y.; YE, W.; HEIDEMANN, J. Energy and latency control in low duty cycle mac protocols. In: *Wireless Communications and Networking Conference, 2005 IEEE*. [S.l.: s.n.], 2005. v. 2, p. 676–682 Vol. 2. ISSN 1525-3511.

LOUREIRO, A. A. F.; NOGUEIRA, J. M. S.; RUIZ, L. B.; MINI, R. A. de F.; NAKAMURA, E. F.; FIGUEIREDO, C. M. S. Redes de sensores sem fio. *Simpósio Brasileiro de Redes de Computadores*, p. 179–226, 2003.

MACEDO, D. F.; CORREIA, L. H. A.; SANTOS, A. L. dos; LOUREIRO, A. A. F.; NOGUEIRA, J. M. S. A rule-based adaptive routing protocol for continuous data dissemination in wsns. *J. Parallel Distrib. Comput.*, Academic Press,

Inc., Orlando, FL, USA, v. 66, n. 4, p. 542–555, abr. 2006. ISSN 0743-7315.

Disponível em: <<http://dx.doi.org/10.1016/j.jpdc.2005.10.006>>.

NOROUZI, A.; ZAIM, A. H. An integrative comparison of energy efficient routing protocols in wireless sensor network. *Wireless Sensor Network*, v. 4, p. 65–75, 2012.

POLASTRE, J.; HILL, J.; CULLER, D. Versatile low power media access for wireless sensor networks. *SenSys'04*, v. 4, 2004.

SRIVASTAVA, V.; MOTANI, M. Cross-layer design: a survey and the road ahead. *Comm. Mag.*, IEEE Press, Piscataway, NJ, USA, v. 43, n. 12, p. 112–119, dez. 2005. ISSN 0163-6804. Disponível em: <<http://dx.doi.org/10.1109/MCOM-2005.1561928>>.

TANENBAUM, A. S. *Redes de Computadores*. [S.l.]: Editora Campus, 2003. 940 p.

TAVARES, J.; VELEZ, F.; FERRO, J. Application of wireless sensor networks to automobiles. *Measurement Science Review*, v. 8, p. 65–70, 2008.

TORRE, P. Padilla de la; PAEZ, J. C.; FERNANDEZ, G. M.; VERDEJO, J. E. D.; TEODORO, P. G.; GOMEZ-CALERO, C. On the influence of the propagation channel in the performance of energy-efficient geographic routing algorithms for wireless sensor networks (wsn). *Wireless Personal Communications*, online, p. 24, 2012.

YE, W.; HEIDEMANN, J.; ESTRIN, D. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM TRANSACTIONS ON NETWORKING*, v. 12, 2004.

YICK, J.; MUKHERJEE, B.; GHOSAL, D. Wireless sensor network survey.  
*Computer Network*, v. 52, p. 2292–2330, 2008.

YU, Y.; GOVINDAN, R.; ESTRIN, D. *Geographical and Energy Aware Routing:  
a recursive data dissemination protocol for wireless sensor networks*. [S.l.], 2001.