

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Informática

Um Protocolo de Reposição de Sensores via Robôs
em Redes de Sensores Sem Fio

Larissa Lucena Vasconcelos

Dissertação submetida à Coordenação do Curso de Pós-Graduação em
Ciência da Computação da Universidade Federal de Campina Grande -
Campus I como parte dos requisitos necessários para obtenção do grau
de Mestre em Ciência da Computação.

Área de Concentração: Ciência da Computação

Linha de Pesquisa: Redes de Computadores e Sistemas Distribuídos

Marco Aurélio Spohn

(Orientador)

Campina Grande, Paraíba, Brasil

©Larissa Lucena Vasconcelos, Agosto de 2010

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

V329p Vasconcelos, Larissa Lucena

Um protocolo de reposição de sensores via robôs em redes de sensores sem fio / Larissa Lucena Vasconcelos. — Campina Grande, 2010.

63 f. : il. color.

Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática.

Referências.

Orientador: Prof. PhD. Marco Aurélio Spohn.

1. Redes de Sensores sem Fio. 2. Reposição de Sensores. I.
Título.

CDU – 004.772.4 (043)

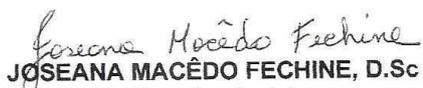
"UM PROTOCOLO DE REPOSIÇÃO DE SENSORES VIA ROBÔS EM REDES DE SENSORES SEM FIO "

LARISSA LUCENA VASCONCELOS

DISSERTAÇÃO APROVADA EM 20.08.2010



MARCO AURELIO SPOHN, Ph.D
Orientador(a)



JOSEANA MACÊDO FECHINE, D.Sc
Examinador(a)



FABIANO SALVADORI, D.Sc
Examinador(a)

CAMPINA GRANDE - PB

Resumo

Redes de sensores sem fio são redes muito suscetíveis a falhas de nós sensores que comprometem a cobertura e a conectividade da rede e, conseqüentemente, o funcionamento desejado da mesma. Normalmente, a manutenção física dos sensores é inviável, sendo então necessário um mecanismo de reposição de nós, *redployment*, que recupere a qualidade de serviço desejada da rede, aumentando assim o tempo de vida da mesma. Este trabalho propõe um protocolo de *redployment* que utiliza robôs dotados de mecanismo de comunicação e de mobilidade para repor os sensores na rede segundo demanda. Alguns sensores da rede são responsáveis por detectarem falhas de uma determinada área e requisitar a um robô que venha repor nós para que o problema no ponto da falha seja sanado. Também o protocolo utiliza uma estratégia, chamada *broadcast* controlado, baseada no conceito de *quorum*, que reduz consideravelmente o tráfego de mensagens na rede, fator importante para ajudar os nós a economizarem energia. Para a avaliação do desempenho do protocolo foram utilizadas simulações. Os resultados mostram que o protocolo é eficiente, conseguindo reparar um problema de cobertura e/ou conectividade em poucos minutos e, ainda, mantendo a rede com nível de qualidade de serviço desejado durante grande parte do tempo das simulações.

Abstract

Wireless sensor networks are very susceptible to sensor failures that may endanger the network coverage and connectivity and, thus, its desired operation. Since it may not be feasible to physically fix the sensors, a mechanism of sensors replacement, called redeployment, is required to recover the network coverage, increasing its life-time. This work proposes a redeployment protocol that uses robots equipped with communication and mobility mechanisms to reset sensors in the network according to demand. Some network sensors are responsible for detecting faults in a given area and request a robot to restore nodes, so that, the problem at the point of failure can be remedied. Also, the protocol uses a strategy, called controlled broadcast, based on the quorum concept, which reduces the message traffic on the network, an important factor in helping nodes saving energy. Simulations were the instrument used to analyze the protocol's performance. The results of these simulations show that the protocol is efficient, managing to fix a problem of coverage and/or connectivity in a few minutes, and even keeping the network quality of service required level for most of the time of the simulations.

Agradecimentos

Primeiramente, agradeço a Deus por todos os motivos, desde a minha existência. Depois, a Elza, minha mãe, meu tudo, meu norte, meu espelho, meu exemplo. A minha irmã, Eveline, que mesmo estando distante durante todo o meu mestrado, sempre me apóia.

Agradecimento também muito especial vai para uma pessoa que realmente me deu seus dois ombros por inteiro no momento mais delicado dessa jornada, meu namorado Alvaro. Com sua paciência, calma, dedicação e carinho, foi um conforto e um porto seguro.

Um grande obrigada dedicado ao meu orientador, Marco Spohn, por ter me acompanhado de maneira exemplar, sempre preocupado, paciente, com sua inteligência e conhecimento ímpar. Sem dúvida um grande exemplo de profissional a ser seguido.

Agradeço a CAPES, pelo apoio financeiro a este trabalho.

Agradeço agora aos familiares próximos, alguns em especial, como Dona Dudu, minha querida, forte e destemida avó; meus tios e tias, Elzanir, Edna, Walter, Iraguacy e Ventura; e primos, Alana, Aline, Rodolfo e Danilo. Familiares estes que, cada um à sua maneira, sempre zelaram por mim e, mesmo indiretamente, sempre fizeram essa jornada se tornar mais fácil.

Por esse mesmo motivo, agradeço aos meus estimados amigos, que não vou listar todos se não seriam páginas e páginas. Alguns dos mais presentes são Julianna, Lorena, Renata, Diego, Mariana, Anderson, Joseana, Neto e Mirna. Obrigada por fazerem minha vida mais colorida, mais leve.

Agradeço ao NS-2, por ter sido, não uma pedra, mas uma montanha no meu caminho, mas que me fez perceber que com determinação, qualquer pessoa pode se tornar um alpinista profissional, vencendo as dificuldades até chegar ao cume da montanha.

Por último, mas não menos importante, agradeço a São Longuinho, por ter me ajudado tantas vezes, durante este trabalho, a encontrar uma saída quando parecia que nada do que eu fazia resolvia meus problemas. Talvez eu ainda deva uns pulinhos das promessas feitas a ele, perdi as contas. Mas vou pagar, São Longuinho, vou pagar!

Conteúdo

| | | |
|----------|--|-----------|
| 1 | Introdução | 1 |
| 1.1 | Motivações | 2 |
| 1.2 | Objetivos | 3 |
| 1.3 | Relevância | 4 |
| 1.4 | Estrutura da Dissertação | 5 |
| 2 | Redes de Sensores Sem Fio | 6 |
| 2.1 | <i>Deployment</i> | 13 |
| 2.2 | <i>Redeployment</i> | 17 |
| 3 | <i>Sensor Redeployment via Robots Protocol</i> | 20 |
| 3.1 | <i>Broadcast</i> Controlado | 21 |
| 3.1.1 | Protocolo de Roteamento | 23 |
| 3.2 | Funcionamento do SRRP | 24 |
| 3.2.1 | Pacotes | 25 |
| 3.2.2 | Eventos do Protocolo | 27 |
| 4 | Apresentação e Análise dos Resultados | 36 |
| 4.1 | Ambiente das simulações | 36 |
| 4.2 | Experimentos | 40 |
| 4.2.1 | Variando o alcance do rádio de transmissão/recepção dos nós | 40 |
| 4.2.2 | Variando a velocidade dos robôs | 43 |
| 4.2.3 | Variando o número de robôs | 48 |
| 4.2.4 | Variando os pesos dos fatores de decisão do coeficiente de atendimento | 50 |
| 4.2.5 | Experimento otimizado | 53 |

5 Considerações Finais

56

Lista de Siglas e Abreviaturas

SRRP - *Sensor Redeployment via Robots Protocol*

AODV - *Ad hoc On Demand Distance Vector*

RREQ - *Route Request*

RREP - *Route Reply*

GPS - *Global Positioning System*

ABROAD - *Announce Broadcast*

RBROAD - *Request Broadcast*

DCA - *Distributed Clustering Algorithm*

DMAC - *Distributed and Mobility Adaptive Clustering*

IP - *Internet Protocol*

CA - *Coeficiente de Atendimento*

Lista de Figuras

| | | |
|------|---|----|
| 2.1 | Arquitetura de um sensor inteligente. | 7 |
| 2.2 | Arquitetura de uma Rede de Sensores sem Fio. | 11 |
| 3.1 | Exemplos de <i>quorums</i> (anúncios e pedidos). | 22 |
| 3.2 | Funcionamento dos <i>quorums</i> | 23 |
| 3.3 | Estrutura das mensagens <i>request</i> , <i>announce</i> e <i>moveRequest</i> | 25 |
| 3.4 | Estrutura da mensagem <i>mediatorResponse</i> | 26 |
| 3.5 | Estrutura das mensagens <i>infoRequest</i> , <i>moveAck</i> e <i>destinationReachedAck</i> | 26 |
| 3.6 | Estrutura da mensagem <i>infoResponse</i> | 27 |
| 3.7 | Envio de <i>announces</i> | 28 |
| 3.8 | Envio de <i>infoRequest</i> | 29 |
| 3.9 | Envio de <i>request</i> | 30 |
| 3.10 | Envio de <i>mediatorResponse</i> | 31 |
| 3.11 | Envio de <i>infoResponse</i> | 32 |
| 3.12 | Envio de <i>moveRequest</i> | 33 |
| 3.13 | Envio de <i>moveAck</i> | 34 |
| 3.14 | Deslocamento do robô ao ponto de falha. | 34 |
| 3.15 | Envio de <i>destinationReachedAck</i> e <i>announce</i> em novo <i>quorum</i> | 35 |
| 4.1 | Grade representando a área da rede (ainda sem robôs) | 38 |
| 4.2 | Gráfico do alcance do rádio <i>versus</i> a média do tempo de atendimento. | 41 |
| 4.3 | Gráfico do alcance do rádio <i>versus</i> o tempo de rede coberta. | 42 |
| 4.4 | Gráfico da velocidade dos robôs <i>versus</i> a média do tempo de atendimento (rádio de 50m). | 44 |

| | | |
|------|---|----|
| 4.5 | Gráfico da velocidade dos robôs <i>versus</i> o tempo de rede coberta (rádio de 50m). | 44 |
| 4.6 | Gráfico da velocidade dos robôs <i>versus</i> a média do tempo de atendimento (rádio de 75m). | 46 |
| 4.7 | Gráfico da velocidade dos robôs <i>versus</i> o tempo de rede coberta (rádio de 75m). | 47 |
| 4.8 | Gráfico do número de robôs <i>versus</i> a média do tempo de atendimento. | 49 |
| 4.9 | Gráfico do número de robôs <i>versus</i> o tempo de rede coberta. | 49 |
| 4.10 | Gráfico dos pesos dos fatores de decisão <i>versus</i> a média do tempo de atendimento. | 51 |
| 4.11 | Gráfico dos pesos dos fatores de decisão <i>versus</i> o tempo de rede coberta. | 52 |
| 4.12 | Gráfico do cenário otimizado: média do tempo de atendimento. | 54 |
| 4.13 | Gráfico do cenário otimizado: tempo de rede coberta. | 55 |

Lista de Tabelas

| | | |
|-----|---|----|
| 4.1 | Descrição dos cenários. | 39 |
| 4.2 | Ambiente de simulações do SRRP. | 40 |
| 4.3 | Variação do alcance do rádio. | 41 |
| 4.4 | Variação da velocidade dos robôs utilizando rádio de 50m. | 43 |
| 4.5 | Variação da velocidade dos robôs utilizando rádio de 75m. | 46 |
| 4.6 | Variação do número de robôs. | 48 |
| 4.7 | Variação dos pesos dos fatores de decisão. | 51 |
| 4.8 | Resultados do cenário otimizado. | 54 |

Capítulo 1

Introdução

Nos últimos anos, tem-se percebido uma grande utilização de redes de sensores sem fio principalmente como soluções militares, médicas e até para escritórios e casas. Alguns exemplos de aplicação são: controle de temperatura ou luminosidade de um ambiente, monitoração de dados fisiológicos, detecção de incêndios e enchentes, detecção de intrusos, monitoração de animais, vigília de campos de batalha e monitoração de tráfego ajudando a escolher uma melhor rota em determinado momento [12].

Isto se deve, principalmente, ao avanço das tecnologias de *hardware* dos sensores e da comunicação sem fio que culminou num interesse crescente de pesquisas que viabilizassem a implantação e a manutenção de redes de sensores sem fio de forma cada vez mais eficiente e com menor custo [26].

Segundo Tilak *et al.* [30], as redes de sensores sem fio são um novo paradigma de redes que tem por finalidade perceber, coletar e disseminar informações sobre um fenômeno do ambiente em que atua. Normalmente, essas redes são compostas de um grande número de nós distribuídos que possuem a capacidade de se auto-configurar e se adaptar a novas situações [24].

Essas características são importantes neste cenário devido ao fato deste tipo de rede ser muito suscetível a falhas e perdas de nós, não sendo, muitas vezes, possível ou viável o acesso físico aos nós a fim de realizar manutenção, já que a implantação, ou *deployment*¹, da rede pode ter sido executada em ambientes remotos e de difícil acesso.

Essa suscetibilidade a falhas se deve, em grande parte, aos nós sensores serem normal-

¹Implantação da rede. Momento em que é definida a posição dos nós da rede.

mente bem pequenos e sensíveis, podendo sofrer danos físicos de elementos do ambiente em que atuam, como um animal em uma floresta, ou ainda no momento do *deployment*, já que em alguns ambientes de difícil acesso, ele pode ser executado a partir de, por exemplo, um avião soltando randomicamente os nós no ambiente.

Outra característica frequentemente presente nessas redes é o modo cooperativo em que os nós trabalham a fim de realizar as atividades para as quais a rede foi projetada [24]. Ao invés de tentar transmitir os valores de sensoriamento diretamente até a estação-base, um nó os transmite apenas até um nó vizinho, desencadeando um processo de retransmissão a cada salto até que a informação atinja a estação-base. Esse esquema de transmissão permite a redução do gasto de energia, pois o transmissor de um nó não precisa ter uma potência muito alta, apenas o suficiente até alcançar seu vizinho.

Aliás, a quantidade limitada de energia que os nós possuem é uma característica extremamente relevante, impactando em todo o planejamento da rede, desde a concepção ou escolha dos protocolos e algoritmos que serão utilizados, até a forma como o *deployment* será executado [30].

1.1 Motivações

Mesmo sendo utilizadas medidas para aumentar a tolerância a falhas, como grande número de nós redundantes, uso de menor quantidade de energia possível (já que a exaustão de energia é uma falha), etc., ainda assim é frequente a diminuição no número de nós em uma rede de sensores sem fio. Essa redução pode ser ocasionada pelo inevitável esgotamento de bateria, ataques maliciosos à rede, falta de visibilidade de outro nó, impedindo a comunicação, ou até por dano físico dos sensores, seja no momento do *deployment*, ou durante a operação da rede.

As proporções deste problema aumentam bastante se ele ocorrer em uma única área, principalmente se for uma área mais crítica, comprometendo a cobertura e/ou conectividade da rede naquela área.

Cobertura e conectividade são questões diretamente relacionadas à qualidade de serviço provida pela rede. Por cobertura, entende-se o quão bem a rede está conseguindo observar os eventos de determinada região, considerando a confiança e a rapidez na coleta das infor-

mações [26; 13]. Já a questão da conectividade está relacionada ao fato de verificar se as informações colhidas estão conseguindo ser transmitidas pela rede de sensores até a estação-base considerando também o tempo de transmissão dessas informações e o custo de energia [35].

Ao ocorrer uma diminuição da qualidade de serviço, não é desejável que a rede continue a operar dessa forma, sendo necessário algum método de recuperação da mesma, para que mantenha a rapidez e exatidão na observação dos eventos críticos.

Como já explicitado, na maioria das vezes não é viável realizar manutenção física de nós sensores, fazendo consertos ou recarregando suas baterias, pois chegar até eles é uma tarefa árdua e muito custosa. Uma solução utilizada para recuperar a cobertura e conectividade da rede é fazer novo *deployment*, ou *redemption*², de nós na rede.

Um aspecto importante a ser considerado é a velocidade de recuperação da qualidade de serviço da rede, pois é desejável que o processo de recuperação se dê rapidamente, para evitar que vários eventos importantes deixem de ser observados e/ou informados.

É importante considerar também que se for fisicamente inviável fazer *deployment* e/ou manutenção de sensores no ambiente, igualmente inviável será fazer fisicamente um *redemption*. Portanto, faz-se necessário o desenvolvimento de soluções de *redemption* que não exijam o acesso ao ambiente, como colocar, durante o *deployment*, sensores móveis que podem migrar para áreas com posteriores problemas de qualidade de serviço, ou robôs móveis que carregam sensores e os levam para áreas com problemas durante a execução da rede.

1.2 Objetivos

Com base nas discussões anteriores, o objetivo deste trabalho é desenvolver uma solução de *redemption* eficiente de nós em uma rede de sensores sem fio sem a necessidade de acesso físico humano à área da rede, tendo a finalidade de manter um nível satisfatório de qualidade de serviço na rede.

Dado que se conhece uma ou mais áreas da rede em que a qualidade de serviço já não é satisfatória, a idéia é recobrar esse nível de qualidade rapidamente, para minimizar o número

²Realocação ou reposição de nós após o início da execução da rede.

de eventos ocorridos na área monitorada que a rede não tenha percebido e/ou reportado.

Para tanto, são utilizados robôs que carregam sensores e se deslocam, segundo demanda, aos pontos com problemas de cobertura e/ou conectividade, deixando em cada ponto um sensor que se integrará à rede com a finalidade de recobrar a qualidade de serviço.

Para analisar a eficiência da solução desenvolvida, foram feitas simulações usando o *Network Simulator – NS-2* [15], das quais foram extraídas as seguintes métricas:

- Tempo médio de atendimento a uma requisição. O tempo de atendimento a uma requisição é contado a partir do momento exato da requisição até o momento em que o robô entrega o sensor no ponto de falha;
- Porcentagem de tempo de rede coberta e conectada. Porcentagem do tempo total da simulação em que a rede estava coberta e conectada; ou seja, em que não havia nenhum pedido de atendimento.

1.3 Relevância

Redes de sensores sem fio são amplamente utilizadas para monitorar eventos críticos como o início de um incêndio ou um ataque do exército inimigo [36].

Para esse tipo de aplicação, de nenhum ou pouco valor é a chegada de uma informação errônea ou tardia sobre um evento, ou ainda a falta de capacidade de observar um evento importante. Por exemplo, não é interessante ter uma rede que detecta de forma recorrente um falso alarme de incêndio; ou que detecta o início de incêndio, mas que só consegue informar sobre esse evento quando o incêndio já tomou maiores proporções; ou, pior ainda, que não detecte um início de incêndio [26].

Então, é de grande importância que a rede se mantenha sempre operando com um bom nível de qualidade de serviço, mantendo a rapidez e a exatidão na observação dos eventos críticos.

No entanto, como já discutido anteriormente, as redes de sensores sem fio são muito suscetíveis a falhas, sejam elas por danos, ataques, falta de alcance a um vizinho ou exaustão de energia [24].

Essa susceptibilidade acaba resultando em uma grande probabilidade da cobertura e conectividade da rede serem prejudicadas em um ou vários pontos da mesma em algum momento de sua operação.

Diminuindo a qualidade de serviço, a rede não opera de modo esperado, estado este que não interessa a quem a implantou. Faz-se, então, primordial o uso de algum mecanismo para sua recuperação.

Nos casos em que não é viável fazer a manutenção física dos sensores da rede, ou a reposição de nós nas áreas em que a qualidade de serviço já não satisfaz, uma solução possível é repor nós sem precisar realizar novo acesso físico à área da rede.

Para isso, pode-se utilizar robôs que são capazes de fazer o *redployment* e são colocados na área da rede durante a sua implantação, conseguindo assim, uma maneira de prolongar o tempo de vida da rede. Tempo este que vai depender agora não apenas dos sensores já implantados, mas também de fatores como o número de sensores que os robôs possuem, a quantidade de energia versus o consumo do robô e quantidade de robôs na rede.

1.4 Estrutura da Dissertação

O restante do trabalho está organizado da seguinte maneira: no Capítulo 2 são tratadas as principais características das redes de sensores sem fio, sendo também citados alguns trabalhos relevantes de *deployment* e *redployment* nesse tipo de rede. No Capítulo 3, é apresentada a solução desenvolvida. Já no Capítulo 4, são apresentados e discutidos os resultados obtidos em experimentos de avaliação. Finalmente, no Capítulo 5, são feitas as considerações finais, além de serem apontadas dificuldades encontradas no desenvolvimento desse trabalho e sugestões para trabalhos futuros.

Capítulo 2

Redes de Sensores Sem Fio

Avanços nas tecnologias de *hardware* de sensores e de comunicação sem fio nos últimos anos possibilitaram a aplicação de redes de pequenos sensores a fim de monitorar eventos. Esta área, a das redes de sensores sem fio, está sendo cada vez mais explorada e está atraindo grandes esforços em pesquisas no intuito de melhorar e baratear essa tecnologia [24; 39].

Tilak *et al.* [30] define redes de sensores sem fio como sendo “redes formadas de pequenos dispositivos de sensoriamento que são implantados de maneira *ad hoc* e que monitoram um fenômeno físico em cooperação”. Estes dispositivos transformam os eventos observados em dados e os transmitem de forma cooperativa para uma estação-base, devendo, assim, ser providos de mecanismos sensoriais, de computação e de comunicação sem fio [3].

Os componentes de *hardware* dos sensores de uma rede são escolhidos dependendo da aplicação da rede, mas, basicamente, os sensores são formados por cinco componentes (Figura 2.1): um controlador, que processa os dados relevantes; memória, para armazenar os programas em execução e dados intermediários; sensores e/ou atuadores, partes do dispositivo responsáveis pela interação com o ambiente físico, observando-o ou controlando-o; dispositivo de comunicação, para que informações possam ser enviadas e recebidas; e suprimento de energia, formado por baterias que alimentam todo o sensor [22].

Pode-se perceber que não há sentido em existir uma rede de sensores se não houver a quem reportar os dados mensurados. Portanto, há uma arquitetura de comunicação das redes de sensores sem fio, formada principalmente por quatro elementos: os próprios nós sensores, que fazem o sensoriamento e reportam as informações para uma estação-base (ou sorvedouro); uma ou mais estações-base (o número vai depender da rede), que se comu-

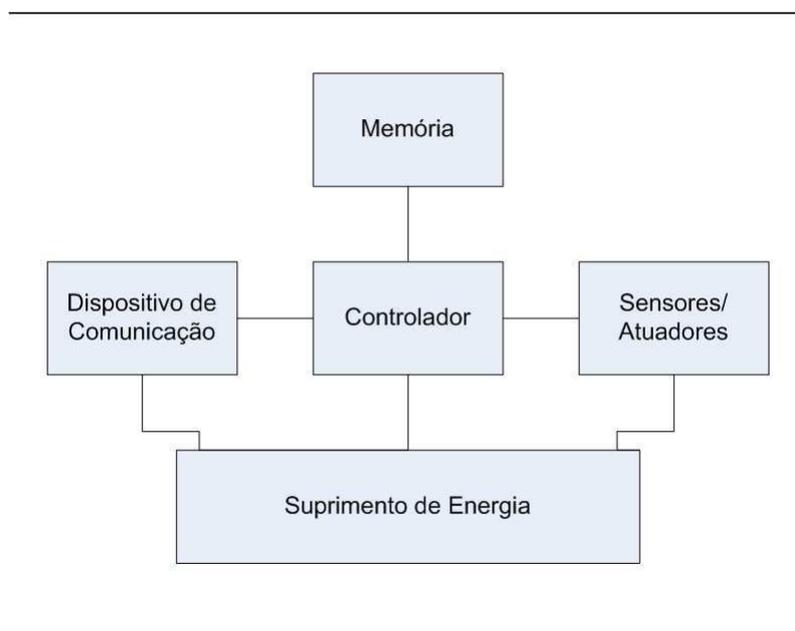


Figura 2.1: Arquitetura de um sensor inteligente.

nicam com o usuário através de uma rede pública ou privada, informando-o os dados que receberam dos sensores; o fenômeno ou evento, ou seja, o acontecimento do qual a rede tem por propósito obter informações; e, por fim, o usuário, o real interessado no sensoriamento dos eventos [3].

As redes de sensores sem fio são orientadas a um propósito específico e, dependendo dele, o modo de transmissão dos dados pode ser diferente, como: [36; 22; 40]

1. Transmissão orientada à ocorrência de eventos: a transmissão de informações é realizada apenas quando um determinado evento ocorre.
2. Transmissão periódica de medições: os sensores transmitem os resultados de suas mensurações periodicamente. Podem também transmitir as informações, caso algum evento ocorra no intervalo do período.

Também dependendo da aplicação, as redes de sensores sem fio podem ser implantadas com uma escala de poucas unidades até milhares de nós sensores.

Uma rede de sensores sem fio pode ser caracterizada como homogênea ou heterogênea, dependendo da dimensão, funcionalidade e tipos dos sensores de seus nós. Por exemplo, se uma rede precisa captar imagem e som, todos os nós podem ter os dois sensores, sendo

uma rede homogênea, ou alguns podem possuir o sensor de imagem e outros, o de som, caracterizando uma rede heterogênea em relação ao tipo e funcionalidade dos nós. Se os sensores de imagem precisarem fazer processamento dos dados que coletam, a rede pode ser heterogênea também quanto à dimensão, pois o sensor de imagem, muito provavelmente, será maior por necessitar de um maior poder de processamento para realizar sua tarefa [24].

Sendo assim, uma rede homogênea remete a um *deployment* simultâneo de um grande número de sensores idênticos, não havendo a preocupação com qual sensor deve ser colocado em determinado local, já que todos desempenham as mesmas funções. Em contrapartida, o *deployment* de uma rede heterogênea deve levar em consideração a posição dos sensores, pois eles desempenham papéis diferentes. Considerando o exemplo da rede heterogênea com sensores diferentes para captar imagem e som, mas que deve perceber imagem e som do mesmo evento, deve ser observada, momento do *deployment*, a cobertura da rede em relação aos dois tipos de dados que ela deve sensorear, não podendo haver uma dada área com vários sensores de imagem e poucos ou nenhum de som [10].

As redes de sensores sem fio são, hoje, soluções para diversos campos de aplicação. Alguns exemplos são citados a seguir [30; 24; 36; 22; 25; 37]:

1. Monitoração de tráfego. Podendo informar rotas com menos fluxo ou até coletar dados para fazer estudos de engenharia de trânsito, tentando melhorar o trânsito da área monitorada.
2. Segurança. Detecção de intrusos em empresas, residências, escritórios, campos de batalha. Uma rede de sensores pode ser implantada para permitir o acesso a determinado prédio ou sala de uma empresa sem o uso de chaves. Cada pessoa da empresa carregaria consigo um sensor que a autenticaria e liberaria ou não seu acesso. Além disso, essa mesma rede poderia ser usada para traçar os trajetos de cada pessoa, podendo ser usada para detectar comportamentos fora do padrão ou do permitido.
3. Automação industrial. Podem ser inseridos sensores em algumas peças para fazer testes em linhas de automação industrial, visando descobrir onde está ocorrendo uma montagem errada, ou um gargalo de desempenho.
4. Manutenção preventiva de máquinas. Em uma grande indústria, podem ser acoplados sensores a cada máquina que vão conseguir perceber quando a operação da máquina

sair do padrão com, por exemplo, vibração excessiva. Esses sensores formariam uma rede de maneira a informar ao mantenedor das máquinas esse problema.

5. Agricultura. Uma rede de sensores sem fio pode ser utilizada para monitorar e até controlar os mecanismos de irrigação, fertilização e controle de pestes para que trabalhem de modo exato, dependendo da situação do plantio.
6. Pecuária. Cada animal pode carregar um sensor que mede alguns aspectos que podem ser indícios de problemas de saúde, como a temperatura ou o número de passos dados, e remetem esse problema se a medição ultrapassar um determinado patamar.
7. Monitoração de ambientes. Sejam eles internos (e.g., escritórios, residências) ou externos (e.g., florestas, desertos, oceanos), podendo medir luminosidade, temperatura, umidade do ar, volume de gases poluentes presentes no ar, etc. Essas medições podem ser usadas apenas para controle do ambiente ou relato de informações sobre o mesmo, ou também para já acionar sistemas dependendo dos valores medidos, como um alarme de incêndio. O uso dessas redes nos chamados “edifícios inteligentes” permite que sejam monitoradas a temperatura, luminosidade e umidade (entre outros aspectos) do prédio como um todo não só visando tornar os ambientes mais confortáveis e uniformes, como também focando na redução do consumo de energia.
8. Logística. Pacotes para entrega podem ser equipados com sensores que, ao passar em determinados pontos de checagem instalados pela empresa, avisam que o pacote conseguiu chegar até aquele ponto, facilitando, assim, o rastreamento das entregas, deixando-as ainda mais confiáveis. Também facilita a conferência de entregas mais volumosas, já que é só o receptor requisitar uma resposta de cada sensor, percebendo assim, quais os volumes que foram entregues, ao invés de ir conferindo manualmente.
9. Aplicações militares. Utilizadas para detectar a invasão de inimigos, explosão de bombas ou presença de substâncias tóxicas.

Para atender aos requisitos das aplicações, como as exemplificadas acima, as redes de sensores sem fio fazem, em geral, três tipos de tarefas: sensoriamento, processamento de dados localmente e comunicação sem fio [36; 3].

O sensoriamento é a tarefa mais crítica e, normalmente, os nós só conseguem observar eventos que acontecem muito próximos deles, por causa do pequeno alcance dos sensores. Entretanto, as aplicações muitas vezes requerem que vastas áreas sejam observadas por uma rede com uma boa cobertura, fazendo com que o número de sensores numa rede seja alto.

A cobertura pode ser medida em termos quantitativos, como o tamanho da área monitorada, ou em termos qualitativos, como o nível de cobertura de determinada área. Uma área com boa cobertura tem maior confiabilidade na monitoração dos eventos. Ao se planejar uma rede de sensores sem fio, pode-se determinar a cobertura requerida em termos de qual a melhor cobertura possível com determinado número de nós; ou como conseguir determinada cobertura com o menor número de nós possível.

Redundância de nós pode ser utilizada para assegurar a cobertura de uma rede. Para isso, é necessário que seja determinado quais nós estarão funcionando em dado instante, para que a cobertura seja a desejada com o menor número de nós ativos em determinado momento.

Segundo Xu [36], a comunicação sem fio é ordens de magnitude mais cara que o processamento local em se tratando de energia utilizada. Como os microprocessadores evoluíram mais rapidamente que as baterias, pode sair mais barato fazer parte do processamento dos dados coletados localmente e só depois transmiti-los, economizando energia e descongestionando o canal de comunicação, já que os dados enviados já estarão processados, mesmo que parcialmente.

Em redes de sensores sem fio, os dados não são necessariamente transmitidos fim-a-fim. Eles são transmitidos de vizinho em vizinho até alcançar a estação-base, onde serão enviados para o usuário, também chamado de gerenciador de tarefas (veja Figura 2.2).

Esse modo de transmissão diminui o gasto de energia, pois os transmissores dos sensores só precisam ser potentes o suficiente para alcançar um vizinho. E, dependendo da potência dos transmissores dos sensores que foram escolhidos para a rede, já se tem uma idéia de quantos nós devem estar em determinada área e qual a distância máxima entre eles para que a rede fique coberta.

A avaliação dessas informações é imprescindível antes do *deployment* da rede, seja ele feito de maneira controlada, ou seja, sendo implantados os sensores manualmente, ou de maneira aleatória. No primeiro caso, munido dessas informações, sabe-se exatamente quantos sensores devem ser utilizados e onde devem ser postos. No segundo caso, as informações

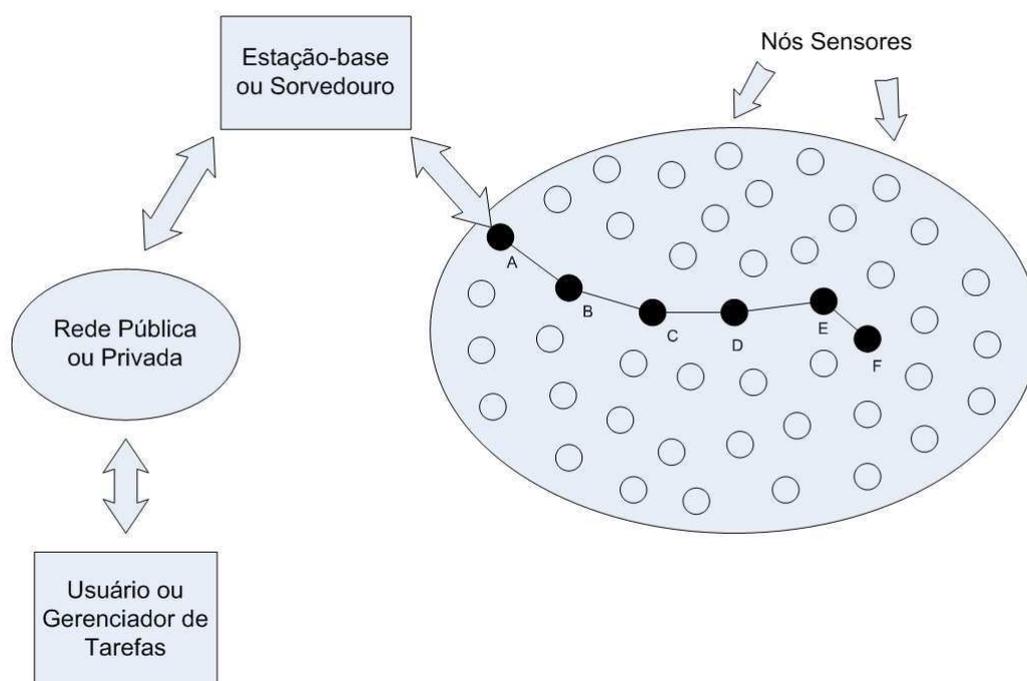


Figura 2.2: Arquitetura de uma Rede de Sensores sem Fio.

podem servir para dar idéia do nível de redundância do número de nós a serem jogados na área, a fim de que haja uma probabilidade ínfima de a rede ficar descoberta já no momento da sua implantação.

Com esse modelo de transmissão, outra preocupação evidente é com a conectividade da rede, pois mesmo que um nó esteja fazendo o sensoriamento corretamente, se ele não conseguir se comunicar com um vizinho, ou se os nós funcionando na rede estão tão disjuntos que os dados percorrem um caminho muito maior que o esperado até a estação-base, de nada vai adiantar essa informação.

Outras características muito importantes estão relacionadas a redes de sensores sem fio e devem ser consideradas no momento do planejamento de uma rede deste tipo. Algumas dessas características são: [30; 22; 40]

1. Tolerância a falhas. Redes de sensores sem fio são muito suscetíveis a falhas de alguns nós por falta de energia, dano físico, ataque ou falta de visibilidade de outro nó na rede. Quando essas falhas não são catastróficas, a rede deve ser capaz de continuar a funcionar de modo que a aplicação não seja prejudicada. Para isso, é usada a já

comentada redundância de nós, fazendo com que uma área seja coberta por outro nó se um deles falhar.

2. Restrições dos nós. Os nós, como já comentado, normalmente são formados por sensor, transmissor/receptor, bateria, memória e processador. Um nó sensor deve ser pequeno, consumir pouca energia, ser barato a ponto de poder ser descartado sem grande impacto no orçamento, ser autônomo e ter a capacidade de se adaptar ao meio em que foi implantado.
3. Latência. Só há interesse em uma informação sobre um evento se ela for recebida com um máximo permitido de atraso, métrica essa dependente de aplicação. Portanto, tanto o sensoriamento e processamento, quanto a disseminação da informação devem ser planejados para atingir um limiar máximo de atraso.
4. Topologia. A topologia de uma rede de sensores sem fio é especialmente importante para seu funcionamento, pois há milhares de sensores que devem funcionar autonomamente, sendo sujeitos a falhas. Os nós são implantados normalmente de forma aleatória, fazendo com que os protocolos utilizados na comunicação sejam auto-organizáveis. Também, durante a operação da rede, a topologia pode mudar drasticamente, seja devido a movimentos dos sensores em redes dinâmicas, ou por falha deles com ou sem reposição, em redes estáticas. Isto faz com que as redes necessitem de protocolos que prevejam essas situações e consigam lidar bem com elas.
5. Escalabilidade. Como já mencionado, uma rede de sensores sem fio pode possuir de poucas unidades a milhares de nós. Além do mais, esse número pode variar no decorrer do tempo. Portanto, os protocolos devem prever e lidar com essa escalabilidade.
6. Confiabilidade. Apenas informações com um nível mínimo de precisão, também dependente de aplicação, são interessantes.
7. Tempo de vida. A rede deve permanecer funcionando como o esperado durante o tempo da função a que se destina ou pelo maior tempo possível, dependendo isso da aplicação. Em contrapartida, ela é formada por sensores que funcionam a baterias que normalmente não podem ser recarregadas, fazendo os sensores inevitavelmente

falharem em algum momento, o que pode diminuir o tempo de vida. Quanto maior a qualidade de serviço, maior consumo de energia, menor o tempo de vida da rede. Há várias maneiras de se medir o tempo de vida de uma rede de sensores sem fio, como o tempo desde o início da atividade da rede até a falha do primeiro nó; ou até a partição da rede em duas ou mais partes, deixando-a desconectada; ou até que uma determinada porcentagem de sensores tenha falhado; ou ainda, até que apareça um buraco de cobertura, ou seja, um determinado ponto da área onde não há mais sensores ativos para fazer o sensoriamento. Aqui, como na tolerância a falhas, a redundância de nós ajuda a diminuir os problemas causados pelo consumo de energia dos nós até a exaustão [27].

8. Programabilidade. Os nós sensores devem ser programáveis para, assim, poder processar os dados que conseguiram perceber, como também para reagir a mudanças em suas tarefas durante a execução da rede.
9. Auto-manutenibilidade. A rede de sensores sem fio muda durante sua execução (diminuem os nós, tarefas são alteradas), como também muda o ambiente em que ela está inserida. Assim, a própria rede deve ter mecanismos para monitorar a própria saúde e, a partir disso, fazer mudanças de parâmetros que permitam que a mesma se adapte ao que está sendo requerido dela.

Para assegurar todas essas características em uma rede de sensores sem fio, um bom plano de *deployment* é essencial. Esse plano consiste em combinar os tipos de dispositivos a serem utilizados e suas posições, pelo menos aproximadas, na rede, a fim de que requisitos da aplicação como, cobertura, conectividade e confiabilidade sejam assegurados com o menor custo possível [36].

2.1 Deployment

Em qualquer sistema de monitoração há a preocupação com o plano de *deployment*, que tem em vista a distribuição mais apropriada dos dispositivos que vão monitorar o ambiente para que a tarefa especificada seja executada plenamente [36].

Um trabalho clássico na área é a definição do Problema da Galeria de Arte [19], que tem como objetivo utilizar o menor número de guardas possível para monitorar um polígono da galeria. Cada ponto da galeria deve ser monitorado por um guarda, e cada guarda tem a capacidade de vigiar tudo que está na sua linha de visão.

Esse problema é uma boa analogia para o problema de planejamento de *deployment* de redes de sensores sem fio, que trata de atender aos requisitos da aplicação (como cobertura e conectividade) com o melhor custo-benefício possível, definindo, para isso, tipos de dispositivos, número e local onde eles irão ser implantados.

Com o planejamento de *deployment*, determinam-se os recursos disponíveis e a disposição desses recursos na configuração do sistema, o que leva à determinação dos limites de desempenho da rede de sensores sem fio.

Por isso, muitos esforços têm sido aplicados a esta área, almejando aumentar o tempo de vida da rede em conjunto com outros estudos que tentam melhorar o desempenho da rede de sensores sem fio durante sua operação.

Ertan e Deliç [17] sugerem algumas medidas de qualidade de *deployment* em termos de segurança, para que se possa decidir se o mesmo atende aos requisitos da aplicação ou se um *redployment* será necessário. Essas medidas são:

1. Área fracamente detectada. A razão entre áreas fracamente detectadas e a área total de monitoramento sinaliza o quão adequado está o número de nós para os propósitos da rede. Quanto maior o valor dessa razão, pior a cobertura. Um limiar dessa razão deve ser definido previamente, para que se possa ter um parâmetro para tomada de decisões;
2. *Redeployment*. Técnicas de processamento de imagem são aplicadas a imagens geradas a partir da medida anterior para que seja possível encontrar partes desconectadas da rede, implicando na necessidade de *redployment* de nós nessas áreas;
3. Lados conectados: é uma medida para determinar se áreas fracamente detectadas são caminho entre pontos de início e fim de comunicação, situação esta que pode gerar problemas na comunicação geral da rede;
4. Probabilidade de detecção de violação: é uma medida que determina a probabilidade de um atacante passar despercebido por um caminho a partir de determinado ponto da rede.

Uma abordagem de otimização de *deployment* é a de Chaudhry *et al.* [12], que, utilizando algoritmos genéticos, propõe uma solução para alocação eficiente de nós, considerando cobertura, tolerância a falhas e consumo mínimo de energia para aplicações em que é possível controlar os locais onde os sensores serão implantados.

Xu [36] considera redes de sensores sem fio com duas camadas de nós, uma de nós sensores e outra de nós retransmissores e define um plano de *deployment* que confere garantia de cobertura à rede pelo modo como executa o *deployment* dos nós sensores, conferindo também conectividade através da distribuição de nós retransmissores de acordo com uma função densidade de nós que é otimizada em relação ao consumo de energia.

Batalin e Sukhatme [5] mostram duas alternativas para solucionar o problema da cobertura no momento do *deployment* em redes de sensores sem fio multi-robôs, considerando uma situação em que todos os sensores foram implantados juntos ou bem próximos na área da rede. Não há nenhuma informação global da localização dos robôs na área (*i.e.* coordenadas geográficas). A alternativa *Informative* se baseia na troca de informações sobre as posições dos robôs, como a localização local real e a localização relativa aos outros robôs que estão interagindo. Quando um robô recebe as mensagens dos outros robôs, decide para onde deve ir, almejando o aumento da cobertura total da rede. Os robôs também conseguem evitar obstáculos, diferenciando-os, ainda, de outros robôs.

Howard *et al.* [20] descrevem um algoritmo incremental e guloso para fazer *deployment* de redes de sensores móveis. Nele, os sensores são implantados um a um, cada nó fazendo uso de informações já colhidas pelos outros sensores já implantados para decidir onde deve se localizar. É um algoritmo de fácil entendimento, mas por ser sequencial, normalmente leva mais tempo para ter a rede totalmente implantada.

Chang e Wang [9] formularam um algoritmo chamado *Self-Deployment by Density Control - SDDC*, que realiza reorganização dos nós de maneira eficaz em redes de sensores sem fio móveis de aplicação em busca e resgate, dado que nenhuma informação de localização global é conhecida. É utilizado o controle de densidade de cada nó (densidade esta dada pelo número de nós pertencentes à área de alcance do rádio do nó dividido pelo valor dessa área) para determinar o *deployment* de outros nós concorrentemente, a fim de obter a melhor cobertura da rede mantendo-a conectada o mais rapidamente possível.

Outro trabalho que utiliza realocação de sensores móveis para melhorar o *deployment*

inicial é o de Yang e Cardei [38], que considera que os nós perto da estação-base consomem mais energia. Depois de realizado um *deployment* randômico, essa solução computa uma densidade de nós desejada não uniforme na rede, evitando que haja buracos de energia perto da estação-base, prolongando assim a vida da rede. Um algoritmo centralizado é utilizado para reposicionar sensores móveis de acordo com a densidade já computada com um custo mínimo. O algoritmo é centralizado, pois considera que cada robô-sensor só pode se mover uma vez, então as posições de todos são computadas ao mesmo tempo para maximizar o resultado para a rede inteira.

Corke *et al.* [16] produziram um helicóptero autônomo para executar o *deployment* de nós de uma rede de sensores sem fio. Uma vez no chão, os nós estabelecem uma rede ad hoc e computam sua conectividade. Se não estiver em um patamar aceitável, é requerido ao helicóptero que faça novos *deployments* onde a rede necessita.

Dois trabalhos de Wang, Cao e La Porta tratam de *deployment* de redes de sensores sem fio. O primeiro [33] trata de redes compostas apenas por sensores móveis e propõe alguns algoritmos distribuídos destinados a controlar os movimentos dos sensores, que são inicialmente colocados de forma aleatória na área da rede, a fim de obter um alto nível de cobertura. Em um dos algoritmos, os sensores migram para onde há buracos de cobertura; em outro, os sensores se movem a fim de se distanciarem uns dos outros, obtendo uma distribuição uniforme na área.

Mesmo podendo ser usados para aumentar a cobertura de uma rede, sensores móveis podem ser caros. Então, o segundo trabalho deles [32], propõe, para redes que possuem nós móveis e estáticos, um algoritmo que tenta balancear o custo dos sensores e a cobertura da rede, guiando os movimentos dos sensores móveis. Neste algoritmo, os sensores estáticos detectam um buraco de cobertura e pedem a movimentação de um robô. Este só atende ao pedido se o buraco de cobertura destino for maior do que a área que ele vai deixar sem cobertura ao sair do ponto onde se encontra. Assim, o robô atende sempre a área com maiores buracos na cobertura, aumentando a área total coberta da rede.

Diante dessa breve revisão de trabalhos relacionados ao *deployment* em redes de sensores sem fio, pode-se perceber o quão importante e diversificada é esta questão para um bom funcionamento de um sistema de redes de sensores sem fio.

Entretanto, mesmo com a execução de um bom *deployment* que deixe a rede coberta e

conectada e com um bom nível de redundância de nós, um fato inevitável a qualquer rede de sensores sem fio é a exaustão de energia dos nós. Se essa exaustão se der quando a missão da rede já tiver sido concluída, não haverá nenhum problema. Porém, as baterias dos robôs podem começar a se exaurir com a rede ainda ativa, o que gera mudanças de topologia da rede, podendo ocasionar buracos na cobertura e perda de conectividade, acarretando em potencial perda de eventos e dados críticos.

Sendo assim, vários esquemas de conservação de energia já foram propostos, usando características que economizam energia, como a alternância de estados dos sensores entre o acordado e dormindo [28; 11]. Também foram propostos esquemas de operação que preservam a conectividade da rede, mas mantendo a consciência do consumo de energia, fazendo um balanço entre as duas métricas [6].

Esquemas como os acima mencionados são, muitas vezes, paliativos, minimizando o gasto de energia, mas apenas postergando a exaustão de energia nos sensores. Uma solução adicional a todos esses esquemas apresentados é fazer o *redeployment* de sensores, substituindo aqueles que se exauriram de energia ou até que deram defeito durante a execução da rede, garantindo a continuidade da operação da mesma [10].

2.2 *Redeployment*

O *redeployment* de sensores em redes de sensores sem fio tem como objetivos prolongar o tempo de vida e recobrar a qualidade de serviço da rede um tempo após o início de execução da mesma, quando os nós já começam a falhar, a partir da reposição ou realocação de nós. No caso da reposição de nós, são implantados novos sensores para substituir os sensores que não estão mais funcionando ou ajudar sensores já implantados a desempenhar sua função. Já no caso da realocação de nós, são utilizados sensores móveis que podem se deslocar para pontos que necessitem mais de sua presença, onde há buraco na cobertura, por exemplo, recobrando a qualidade de serviço no local para onde se locomoveu [7].

Mesmo sendo uma área que começou a ser explorada há poucos anos, já existem alguns trabalhos que têm como foco o *redeployment* de sensores em redes de sensores sem fio. Alguns serão apresentados a seguir.

Li *et al.* [23] utilizam uma abordagem com algoritmos genéticos para determinar conti-

nuamente, depois do *deployment*, em que áreas a rede possui densidade de nós insuficiente para garantir o nível de qualidade de serviço desejado, e daí mover robôs utilizados como nós sensores para estes pontos, a fim de melhorar a cobertura da rede no local.

Wu *et al.*[34] propõem um método de *redeployment* de nós móveis em uma rede de sensores formada por sensores estáticos e móveis. Informações sobre buracos de cobertura, *hot spots*, obstáculos e distância dos limites da rede são coletadas, são atribuídos pesos a cada um desses problemas e, a partir desses pesos, são decididas as áreas que estão mais necessitadas de adição de sensores, movendo sensores para as mesmas.

Um outro trabalho na área de *redeployment* é o de Wang e Ding [31]. Nele, são consideradas redes baseadas em *clusters*, ou seja, grupos de nós próximos um dos outros em que cada grupo possui um nó, chamado de *cluster-head*, responsável por veicular a informação daquele *cluster* até o sorvedouro. A solução utiliza duas heurísticas que tentam encontrar as melhores posições para adicionar sensores de modo a prolongar o tempo de vida da rede o máximo possível. Para tanto, eles consideram como fatores determinantes da decisão a probabilidade de ocorrência de eventos nos pontos da área da rede, pontos esses que consumirão mais energia; e o esquema de roteamento para a entrega de mensagens ao sorvedouro, fator este que pede duas heurísticas diferentes, uma para roteamento estático e outra para roteamento dinâmico.

Sheu *et al.* [29] desenvolveram um robô móvel inteligente dotado de comunicação sem fio e implementaram uma aplicação de *redeployment* de nós usando um algoritmo em que os robôs vão até os sensores que já estão com pouca energia e, automaticamente, coloca novos sensores no local.

Quando um sensor percebe que sua energia está por acabar, avisa a um sorvedouro que, então, pede a um robô que vá socorrer o nó. Os robôs são inicialmente posicionados próximos a um sorvedouro.

Para o deslocamento do robô, ele segue o caminho de roteamento estabelecido previamente entre o sorvedouro e o sensor, recebendo do primeiro a informação de qual o próximo nó do caminho. Daí, o robô se encaminha até esse nó e dele recebe o próximo nó, e assim por diante, seguindo salto a salto, até que chegue ao sensor prestes a falhar. Porém, nenhuma informação sobre a localização dos sensores do caminho é disponibilizada. Sendo assim, o robô é guiado pela força dos sinais que ele recebe dos sensores que formam esse caminho

de roteamento, através da rede; ou seja, se a força do sinal proveniente do próximo sensor do caminho diminui, quer dizer que ele está se distanciando do mesmo. O robô então segue em outra direção, tentando encontrar esse sensor do caminho e repetindo essas ações até que chegue ao seu destino, ou seja, onde se localiza o nó que está prestes a falhar. Um mesmo robô também pode atender a vários pedidos de socorro, atendendo primeiro ao pedido proveniente do sensor localizado mais próximo do sorvedouro e depois aos sucessivamente mais distantes.

Cao *et al.* [7] consideram que uma rede pode ter várias missões concorrentes e propõem uma realocação de sensores móveis redundantes quando há modificações em uma missão ou nas condições da rede (buraco de cobertura, ataques, etc). Para fazer essa realocação são considerados alguns fatores como a complexidade computacional, a distância do movimento, a sobrecarga da comunicação e o impacto do movimento em outras missões que estejam ocorrendo concomitantemente na rede.

Ao invés de mover um sensor diretamente para a área necessitada, o *redeployment* é feito em cascata, movendo vários sensores por pequenas distâncias e, conseqüentemente, em menos tempo, tentando sanar o problema de maneira menos onerosa em termos de consumo total de energia, energia mínima remanescente e tempo de realocação.

Para conseguir descobrir onde há nós redundantes sem grande sobrecarga de mensagens na rede, divide-se a área da mesma em células, cada uma possuindo um líder, e usam o conceito de *quorum* [8; 14; 18]. *Quorum* é um conjunto de células e quaisquer dois *quorums* devem possuir uma célula de interseção.

No trabalho, eles definem dois tipos de *quorums*, os de avisos, formados cada um por uma coluna de células da área da rede; e os de pedidos, formados cada um por uma linha de células da área da rede. O líder de uma célula que possui nós móveis redundantes avisa essa condição, mandando uma mensagem apenas para as células de seu *quorum* (coluna). Quando há necessidade de mais sensores em algum ponto, o líder da célula a que pertence esse ponto envia um pedido de ajuda para as células de seu *quorum* (linha). O líder da célula de interseção recebe as duas mensagens e responde ao pedido informando a localização dos sensores redundantes. A sobrecarga de mensagens circulando na rede na descoberta dos sensores redundantes diminui de $O(N)$ ¹ para $O(\frac{\sqrt{N}}{2})$.

¹Número de células da grade.

Capítulo 3

Sensor Redeployment via Robots Protocol

O *Sensor Redeployment via Robots Protocol* (SRRP) é um protocolo de aplicação destinado a controlar, de modo eficiente, a realização de *redeployment* de nós em uma rede de sensores sem fio sem a necessidade de acesso físico à área da rede, visando assim, manter um nível de qualidade de serviço que atenda aos requisitos do(s) propósito(s) da rede.

Para que seja possível repor nós sem novo acesso à área da rede, são utilizados robôs capazes de se comunicar com os sensores e dotados de mobilidade que são posicionados no momento do *deployment* inicial dos nós sensores. Esses robôs carregam sensores consigo e, sob demanda, os colocam na área da rede para substituir outros nós que estejam prestes a falhar. Os robôs utilizados neste protocolo devem possuir um equipamento que os informa de sua posição via GPS (*Global Positioning System*) para poderem se locomover diretamente ao ponto onde foi detectada a necessidade de novos nós, diferentemente daqueles desenvolvidos e utilizados por Sheu *et al.*[29], em que os nós sensores se locomovem de acordo com a força do sinal proveniente dos sensores que participam do caminho de roteamento entre o sorvedouro e o sensor que está por falhar.

Alguns sensores da rede, chamados de líderes, são responsáveis por pedir ajuda aos robôs. Considera-se que um líder é informado quando um sensor próximo está prestes a falhar, o que pode gerar um buraco na cobertura e também afetar a conectividade, ou quando já há um buraco na cobertura próximo à sua localização e os sensores que falharam naquele ponto não conseguiram avisá-lo de tal situação (no caso de um ataque, por exemplo).

Ao receber essas informações, o líder imediatamente inicia um pedido para que algum robô da rede venha repor os sensores daquele ponto, a fim de recobrar a cobertura necessária.

Assume-se que a rede já possua algum mecanismo de eleição dos líderes, podendo ser baseado em algoritmos de eleição de *clusterheads* já existentes, como o *Distributed Clustering Algorithm* (DCA) ou *Distributed and Mobility Adaptive Clustering* (DMAC) [4].

Também os sensores, sejam líderes ou não, devem ser dotados de equipamentos GPS para que tenham ciência de seu posicionamento e, assim, possam informar ao líder, e este ao robô, a exata posição onde devem ser repostos sensores.

Quando um robô recebe um pedido de ajuda e o atende, ao chegar ao ponto de falha onde instalou um novo nó, envia um anúncio dali, já que ele não volta ao ponto de onde partiu. De sua nova localização, ele só se moverá para atender a um novo pedido.

3.1 Broadcast Controlado

Dado que um líder foi informado da ocorrência de um buraco de cobertura formado ou prestes a sê-lo, ele deve fazer um pedido para que um robô possa repor o sensor que já não funciona mais ou está prestes a ter sua bateria esgotada.

Para iniciar esse processo, o líder deveria fazer um *broadcast* do pedido na rede, até que um ou mais robôs o recebessem e se dispusessem a vir atendê-lo. Só que usar *broadcast* para encontrar os robôs aumentaria significativamente a sobrecarga de mensagens circulando pela rede, se, por exemplo, a rede fosse de médio porte e ocorresse várias falhas dentro de um curto intervalo de tempo; ou mesmo em uma rede de pequeno porte, poderia haver ainda muitas mensagens de pedidos circulando pela rede quando um robô já tivesse se prontificado a atendê-lo.

Então, decidiu-se adaptar a idéia proposta por Cao *et al.* [7] (discutida no capítulo anterior) para descobrir onde há sensores redundantes, utilizando-a para que um líder possa descobrir um ou mais robôs sem ser necessário fazer um *broadcast* na rede inteira.

Para tanto, é considerada uma divisão em grade da área onde a rede está estabelecida e em cada célula da grade haverá um líder, que será responsável por detectar falhas na cobertura daquela célula. É utilizado também o conceito de *quorums* (Figura 3.1), cada um formado por uma linha (*quorum* de anúncios), ou coluna (*quorum* de pedidos).

Cada robô anuncia periodicamente sua posição no seu *quorum* de anúncios, ou seja, a todos os sensores que se localizam nas células da mesma linha que a dele. Da mesma forma,

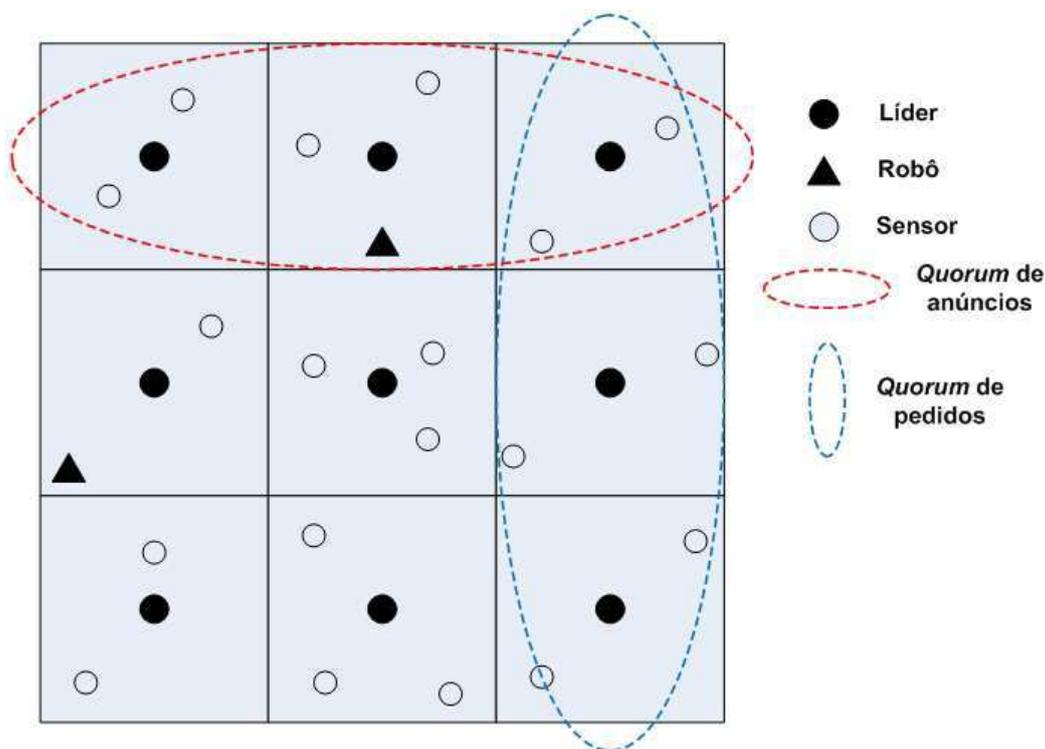


Figura 3.1: Exemplos de *quorums* (anúncios e pedidos).

quando há um pedido a ser feito, um líder o envia ao seu *quorum* de pedidos, ou seja, todos os sensores que se localizam nas células da mesma coluna que a dele recebem o pedido. Essa maneira de fazer anúncios e pedidos é chamada de *broadcast* controlado.

O líder da célula que está na interseção dos dois *quorums*, chamado de mediador, tendo recebido um anúncio e um pedido, responde ao líder emissor do pedido, enviando o endereço IP do robô emissor do anúncio, avisando-o da existência daquele robô (Figura 3.2).

Dessa forma, o protocolo se torna bem mais eficiente sob dois importantes aspectos: a diminuição do número de mensagens circulando na rede e, por consequência desta, a economia do suprimento de energia dos nós (sejam sensores ou robôs), já que estes vão processar e repassar bem menos mensagens (*i.e.*, apenas aquelas enviadas à coluna e linha a que pertencem).

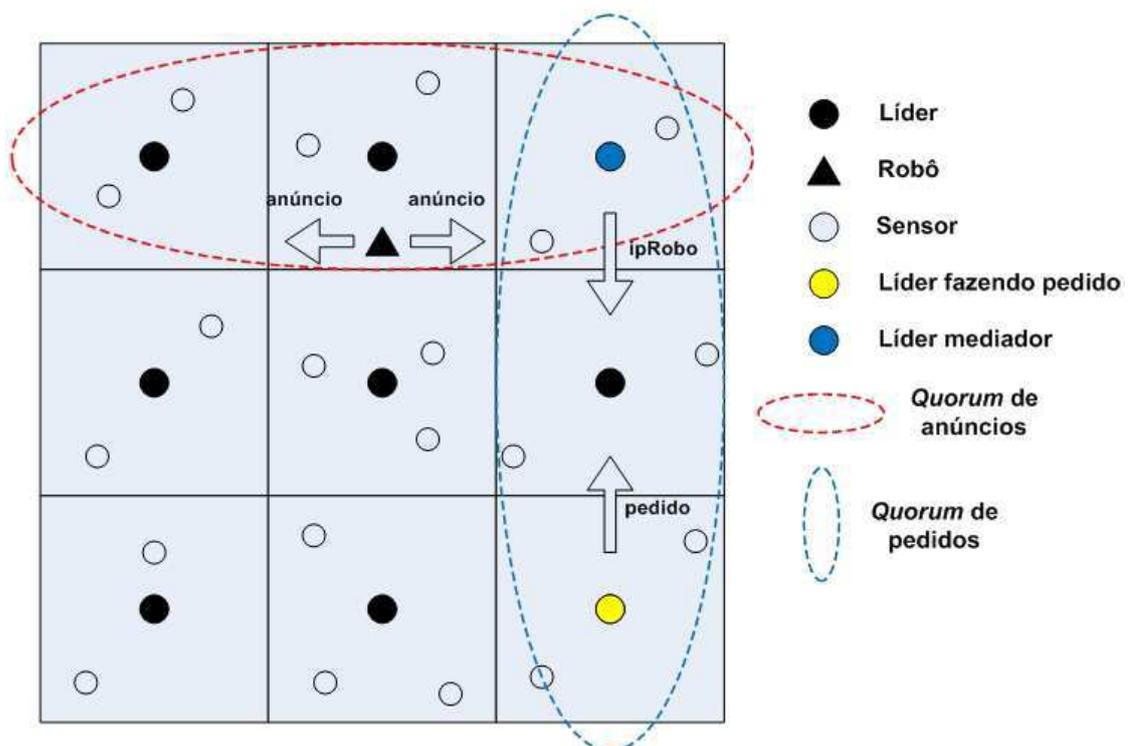


Figura 3.2: Funcionamento dos *quorums*.

3.1.1 Protocolo de Roteamento

Como protocolo de roteamento para ser utilizado com o SRRP, foi escolhido o *Ad hoc On Demand Distance Vector* (AODV) [1], tendo sido necessário fazer algumas adaptações para incorporar o *broadcast* controlado no envio dos anúncios e pedidos.

O AODV é um protocolo de roteamento *unicast* para redes *ad hoc* móveis que apenas estabelece rotas quando são necessárias, ou seja, é reativo. Para isso, um nó envia um pedido de estabelecimento de rota para outro nó (endereçado a ele) via *broadcast* na rede, chamado de *route request* (RREQ). Quando um nó recebe um RREQ e o reencaminha, já guarda em sua tabela de roteamento o caminho inverso para o nó que iniciou o pedido. Quando o nó a que se destina o RREQ o recebe, responde ao nó que o enviou com um *route reply* (RREP). O RREP segue o caminho inverso estabelecido durante o reencaminhamento do RREQ, até o nó que iniciou o pedido de rota, confirmando que esta já foi estabelecida.

A implementação do *broadcast* controlado é parecida com a do RREQ, mas sem o uso de um endereço único de destino a ser atingido pelo pacote. Quando um anúncio ou pedido é

efetuado, um pacote de roteamento endereçado a todos os nós da rede (*broadcast*) é enviado, porém, apenas os nós que pertencem ao mesmo *quorum* processam o pacote; ou seja, o reencaminham e o repassam às camadas superiores, até chegar ao SRRP.

Ao receber um anúncio ou pedido, o nó deve saber se faz parte ou não do *quorum* daquele que enviou o pacote. Sendo assim, o pacote carrega consigo a informação sobre a localização do nó que o envia, sob a forma de coordenadas geográficas. Um nó que recebe esse pacote calcula se está na mesma faixa (linha ou coluna, dependendo se for anúncio ou pedido) do nó que o enviou, o que significa que ele pertence ao referido *quorum*. Em caso afirmativo, o nó processa o pacote.

Quando é enviado um anúncio, o pacote de *broadcast* controlado é chamado de *announce broadcast* (ABROAD), que apenas circula pela mesma linha de células do emissor. Já quando é enviado um pedido, o pacote é chamado de *request broadcast* (RBROAD), que apenas circula pela mesma coluna de células do emissor.

Além disso, esses nós também guardam a rota inversa para quem enviou o ABROAD ou RBROAD, como no encaminhamento do RREQ, pois há a probabilidade de utilização dessa rota em pouco tempo. No caso do ABROAD, um líder que o recebe pode ter a necessidade de enviar um pedido e, assim, já tem a rota para o robô de que recebeu o ABROAD. No caso do RBROAD, um líder que o recebe e tem informação sobre algum robô, já usa essa rota para informar sobre o robô ao líder de que recebeu o RBROAD.

Excetuando-se esses dois tipos de pacotes, o ABROAD e o RBROAD, que são tratados diferentemente pelas razões já explicitadas, todo o resto do roteamento é feito normalmente pelo AODV, sem modificações. Por exemplo, quando um líder recebe de outros líderes informação de robôs disponíveis, já os contacta diretamente enviando RREQs para os endereços dos robôs, caso ainda não possua rota para os mesmos.

3.2 Funcionamento do SRRP

Após ter sido explicitado o modo como é feito o *broadcast* controlado, com modificações no protocolo de roteamento AODV, será apresentado como funciona o SRRP, na camada de aplicação.

3.2.1 Pacotes

Primeiramente, serão apresentadas as mensagens definidas e utilizadas pelo protocolo.

- *Request*

A mensagem *request* é enviada ao *quorum* de pedidos quando um líder vai fazer um pedido de ajuda aos líderes mediadores, objetivando obter informação da existência de robôs.

Essa mensagem é composta pelos campos *tipo*, *coordX* e *coordY* (Figura 3.3). O campo *tipo* denota o tipo da mensagem SRRP, que neste caso tem valor 1, indicando que é um *request*. Os campos *coordX* e *coordY* denotam, respectivamente, as coordenadas geográficas x e y da posição em que se encontra o líder que envia o pedido. Essa localização é utilizada pelo protocolo de roteamento (*i.e.*, o AODV modificado) para que este possa decidir quais nós devem processar essa mensagem por estarem no mesmo *quorum* de pedidos que o líder que a enviou, de acordo com as regras já discutidas do *broadcast* controlado.



Figura 3.3: Estrutura das mensagens *request*, *announce* e *moveRequest*.

- *Announce*

A mensagem *announce* é enviada ao *quorum* de anúncios quando um robô vai fazer um anúncio aos líderes que se localizam na mesma linha da grade.

Da mesma forma que a mensagem anteriormente apresentada, esta é composta pelos campos *tipo*, *coordX* e *coordY* (Figura 3.3). O campo *tipo* agora tem valor 2. Os campos *coordX* e *coordY* denotam, neste caso, a posição do robô na rede, sendo esses campos também usados no momento do *broadcast* controlado para esse tipo de mensagem.

- *MediatorResponse*

Esta mensagem é a resposta de um líder mediador a um *request* de outro líder, informando sobre a existência de um robô que o possa atender. Ela é formada pelos campos *tipo*, com valor 3 neste caso, e *ipRobo* que contém o endereço IP de um robô do qual o mediador recebeu o anúncio (Figura 3.4).



Figura 3.4: Estrutura da mensagem *mediatorResponse*.

- *InfoRequest*

Esta mensagem é enviada por um líder que iniciou um pedido de ajuda a um robô, solicitando a ele informações sobre seu estado atual. É uma mensagem composta apenas pelo campo *tipo*, sendo denotado pelo valor 4 (Figura 3.5).



Figura 3.5: Estrutura das mensagens *infoRequest*, *moveAck* e *destinationReachedAck*.

- *InfoResponse*

Esta é a mensagem enviada por um robô como resposta a um *infoRequest*. Os campos que a compõem são *tipo* (5, neste caso), *coordX* e *coordY*, *energia* e *numSensores*. Como em *announce*, os campos *coordX* e *coordY* denotam a posição atual do robô (Figura 3.11). O campo *energia* contém o valor da quantidade de energia remanescente na bateria do robô. Já *numSensores* informa o número de sensores de que o robô dispõe naquele momento.

Essas informações são utilizadas pelo líder que as recebe para comparar os robôs e decidir qual é melhor para atender a seu pedido.

- *MoveRequest*

| | | | | |
|------------------|---------------------|---------------------|----------------------|--------------------------|
| tipo (1 byte) | coordX (8 bytes) | coordY (8 bytes) | energia (8 bytes) | numSensores (4 bytes) |
|------------------|---------------------|---------------------|----------------------|--------------------------|

Figura 3.6: Estrutura da mensagem *infoResponse*.

Depois de receber *infoResponse* de alguns robôs e decidir qual deles melhor pode lhe atender naquele momento, o líder usa esta mensagem para pedir a um robô que vá até o ponto de falha; ou seja, onde ele deseja que haja reposição de sensores em sua célula. Para isso, ele precisa informar ao robô sobre as coordenadas desse ponto. Portanto essa mensagem é formada, além do campo *tipo* com valor 6, pelos campos *coordX* e *coordY*, contendo a localização para a qual o robô deve se destinar (Figura 3.3).

- *MoveAck*

Um robô ao receber um *moveRequest*, se estiver disponível para atender a este pedido, responde ao líder com um *moveAck* antes de começar sua locomoção ao ponto de falha. Assim como *infoRequest*, esta mensagem é apenas formada pelo campo *tipo*, que recebe o valor 7 (Figura 3.5).

- *DestinationReachedAck*

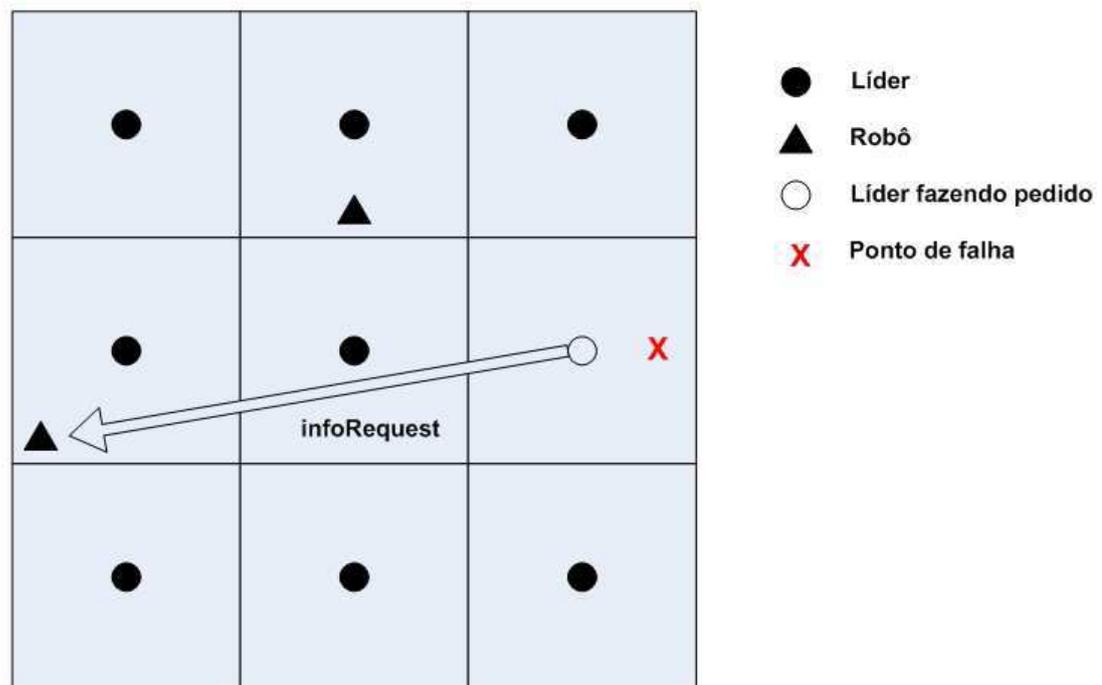
Esta é a mensagem usada por um robô para informar ao líder que chegou ao ponto de falha; ou seja, atendeu a seu pedido. Ela também é composta apenas pelo campo *tipo*, com valor 8, neste caso (Figura 3.5).

3.2.2 Eventos do Protocolo

Anúncios

Assim que a rede começa a executar suas atividades, o SRRP entra em ação, por meio dos robôs que enviam anúncios (mensagens *announce*) periódicos para o seu *quorum* de anúncios utilizando o *broadcast* controlado (Figura 3.7).

Os líderes das células que recebem um (ou vários) desses anúncios, guardam o endereço IP do robô junto com um *tempo de validade do anúncio*. Este tempo de validade é importante

Figura 3.8: Envio de *infoRequest*.

células, os mediadores, podem informá-lo sobre a existência de outros robôs (Figura 3.9). Esse contato com os mediadores é importante, pois:

- quando o líder que faz o pedido ainda não sabe da existência de nenhum robô, perguntar aos mediadores é a única forma de descobrir os IPs dos robôs, dado o uso do *broadcast* controlado;
- mesmo quando o líder já entrou em contato com um ou mais robôs, os mediadores podem informá-lo sobre robôs que estarão em melhores condições de atendê-lo, seja por estarem mais próximos, ou com mais sensores, ou com mais energia em bateria.

Resposta do Mediador

Quando um mediador da mesma coluna onde a falha ocorreu recebe um *request*, responde ao líder com um *mediatorResponse* para cada IP de robô válido que possui em sua lista (Figura 3.10).

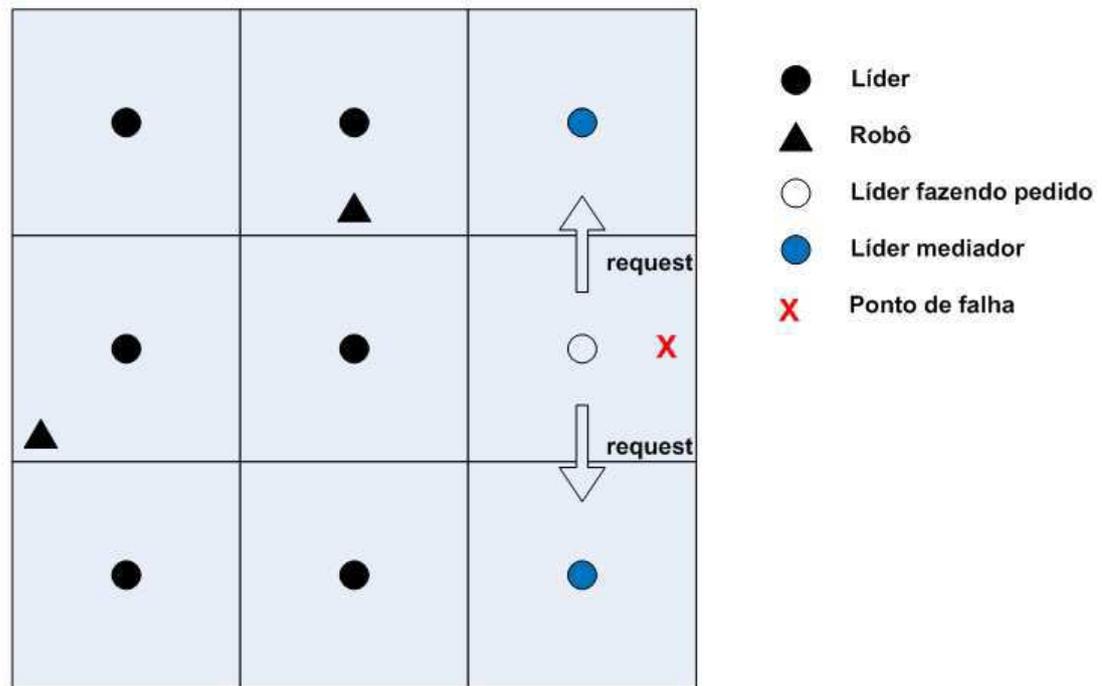


Figura 3.9: Envio de *request*.

Decidiu-se enviar os IPs um a um, pois se fosse enviada toda a lista no mesmo *mediatorResponse* e este não conseguisse chegar ao destino, todos os IPs dessa mensagem iriam se perder. Enviando-os separadamente, a probabilidade de que nenhum chegue ao líder diminui consideravelmente, munindo, assim, o líder de mais possibilidades de robôs para atendê-lo.

Quando o líder recebe os *mediatorResponses*, também entra em contato com os robôs, pedindo seus estados atuais através do envio de *infoRequests*.

Líder recebe respostas dos robôs

Tendo já enviado *infoRequests* a alguns robôs, o líder então recebe resposta de um robô, o *infoResponse* (Figura 3.11). A contar do momento do recebimento dessa resposta, o líder espera pela chegada de até um determinado *número de infoResponses* de diferentes robôs, ou até que o *tempo de espera por infoResponses* expire.

Munido do *número de infoResponses* ou expirado o *tempo de espera por infoResponses* (o evento que ocorrer primeiro), o líder começa a avaliar os estados atuais dos robôs. Para

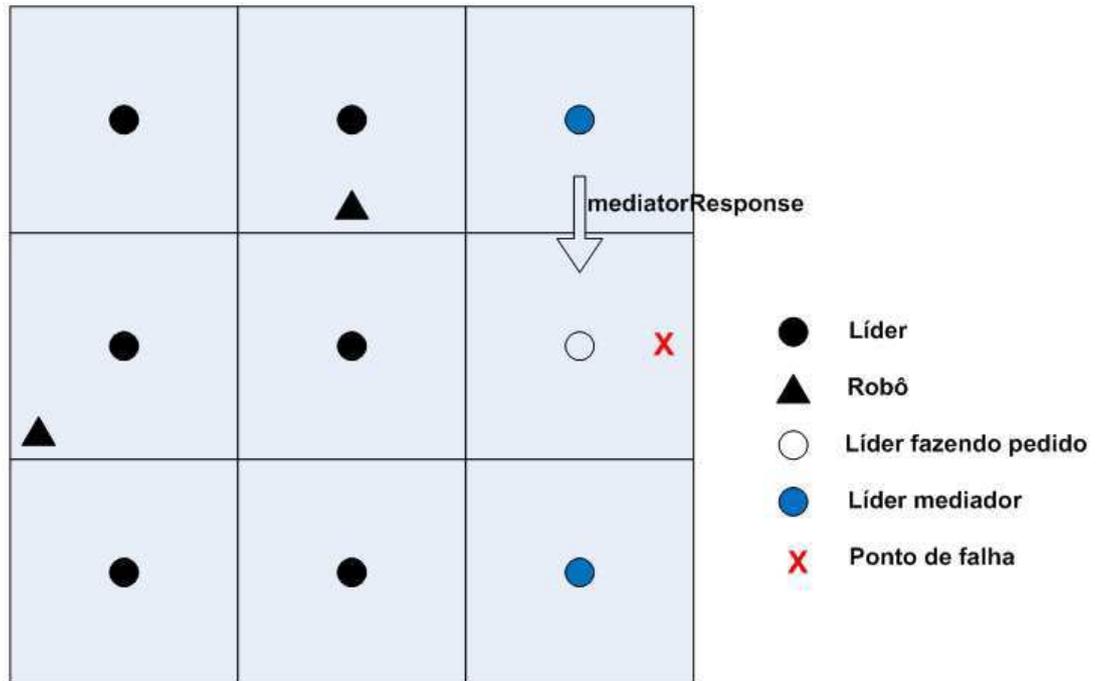


Figura 3.10: Envio de *mediatorResponse*.

cada robô ele calcula:

1. a distância do robô ao ponto de falha, usando para isso as coordenadas recebidas no *infoResponse*;
2. o seu coeficiente de atendimento (CA), dado por:

$$CA = (P_d * \frac{1}{distância}) + (P_e * energia) + (P_{ns} * numSensores)$$

em que P_d , P_e e P_{ns} são pesos atribuídos respectivamente à distância, energia e número de sensores informados pelo robô, de acordo com a importância que se queira dar a cada uma dessas características.

Em seguida, o líder ordena sua lista de robôs de maneira decrescente de acordo com o CA. Assim, a melhor opção de robô que o líder encontrou para atender ao seu pedido (*i.e.*, com maior CA) estará na primeira posição da lista e a pior, na última.

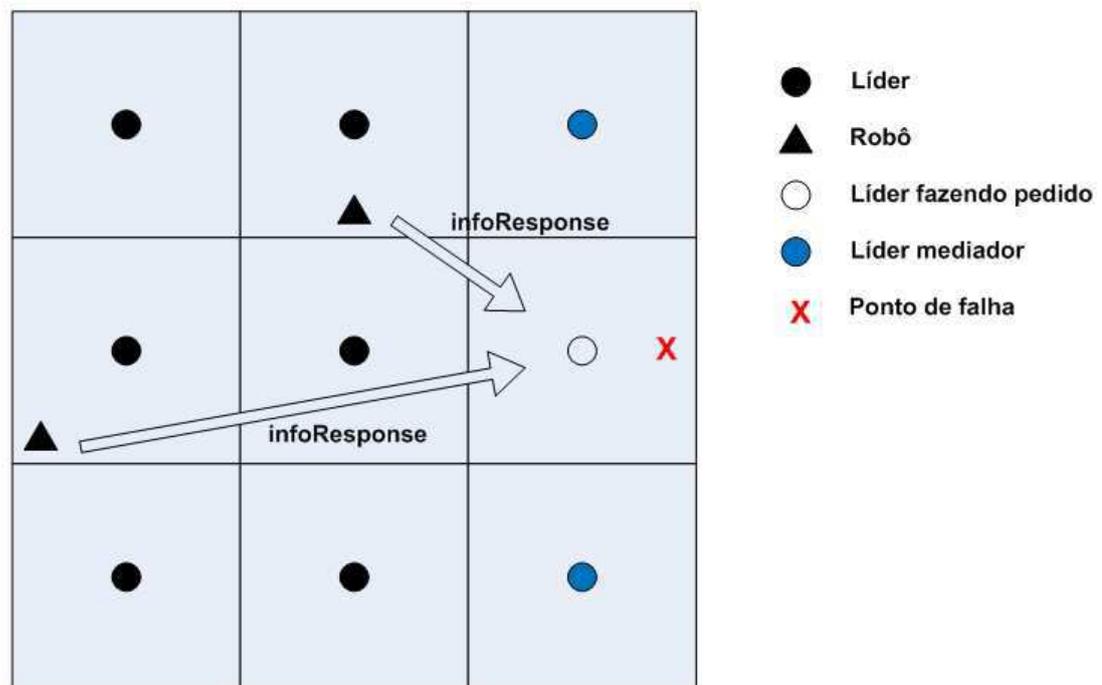


Figura 3.11: Envio de *infoResponse*.

A decisão de esperar pelo *tempo de espera por infoResponses* ou pelo recebimento de determinado *número de infoRequests* se deu para que um líder pudesse dispor de um número considerável de opções de robôs para atendê-lo e não pedir ajuda ao primeiro que o respondesse, sem que, para isso, precisasse esperar muito pelas respostas, o que poderia comprometer o desempenho do protocolo.

Líder solicita atendimento

Já com sua lista de robôs ordenada, o líder envia um *moveRequest* ao primeiro robô da lista (Figura 3.12), solicitando que o mesmo o atenda, informando ao robô o ponto exato da falha (coordenadas (x,y)). Isso para que ele venha colocar um sensor neste ponto, substituindo aquele que está por falhar, ou recuperando a rede do buraco de cobertura já instaurado, recobrando assim a qualidade de serviço local.

O líder então espera, no máximo, por um determinado tempo pelo recebimento de um *moveAck* do robô - o *tempo de espera por moveAck*, ou seja, pela confirmação de que ele vai poder atender à sua solicitação. Caso não a receba, envia o pedido de atendimento ao segundo

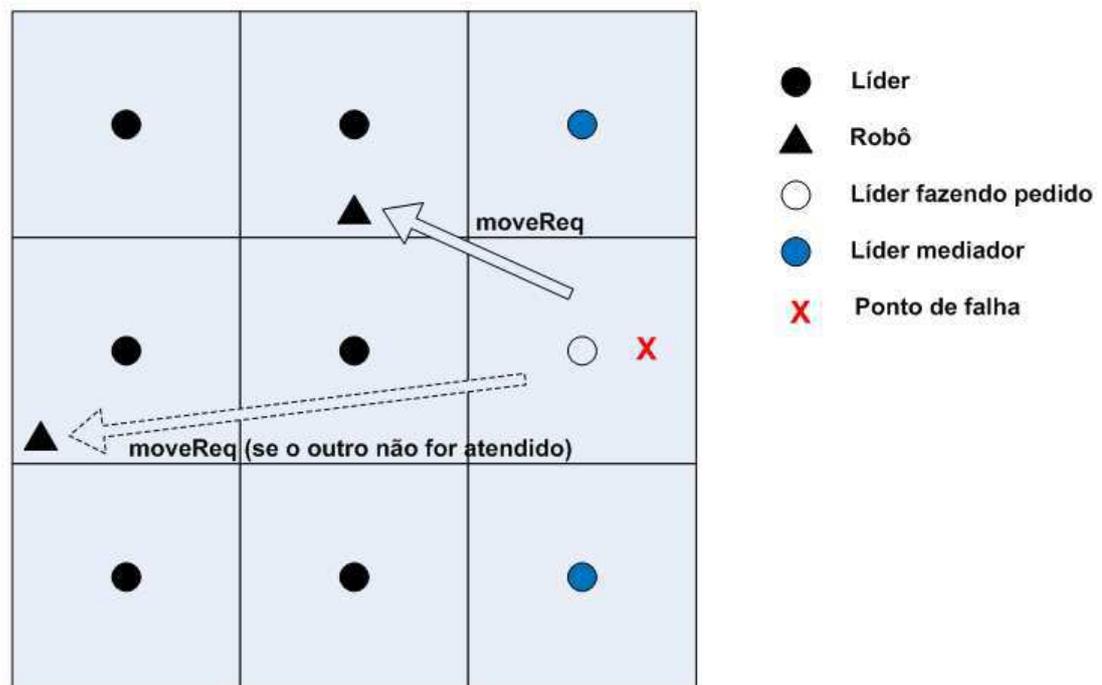


Figura 3.12: Envio de *moveRequest*.

robô da lista e, assim, sucessivamente. Se por algum acaso, não receber nenhuma confirmação, o líder interpreta que nenhum robô da lista vai atendê-lo, e então, inicia novamente um pedido aos mediadores do seu próprio *quorum* de pedidos e envia novos *infoRequests* a outros robôs de que, eventualmente, tenha recebido *announces* durante esse processo.

Robô confirma e inicia atendimento

Quando um robô recebe um *moveRequest*, envia como resposta ao líder um *moveAck* (Figura 3.13); ou seja, a confirmação que pode e vai atendê-lo e, só então, começa a se mover ao ponto de falha (Figura 3.14).

O robô não responde a qualquer outro pedido durante sua movimentação.

Robô confirma sucesso do atendimento

Ao chegar ao ponto em que foi requisitado a depositar um novo sensor, o robô envia ao líder que solicitou sua ajuda um *destinationReachedAck*, confirmando o sucesso do atendimento e,

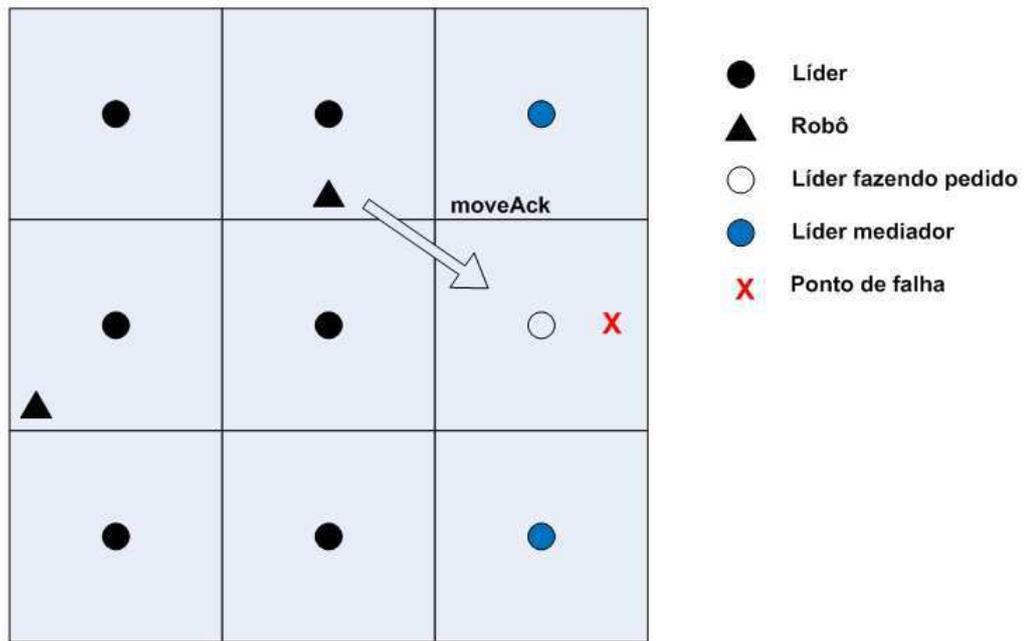
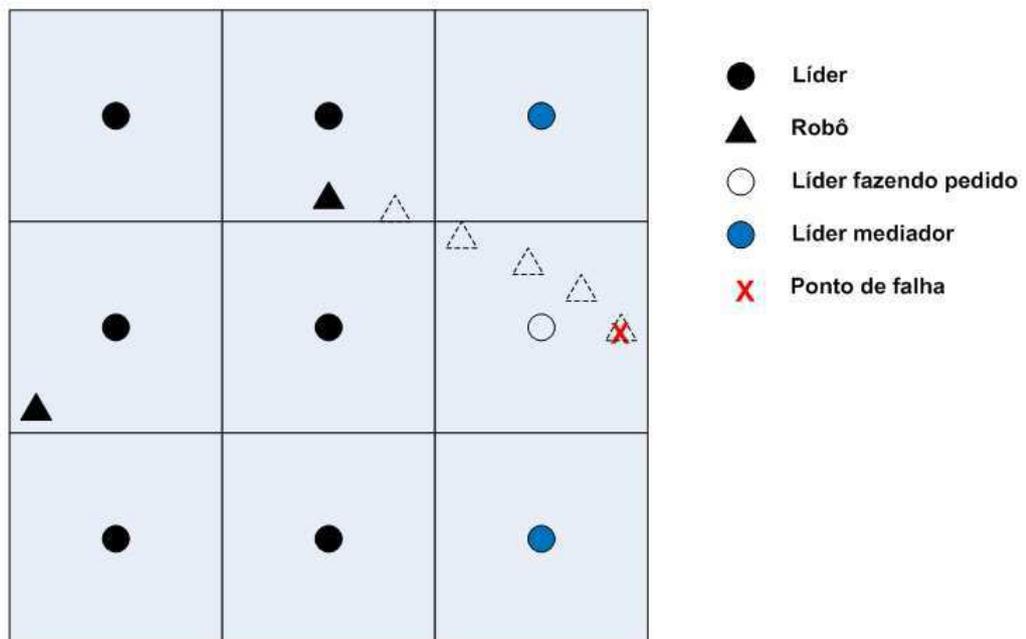
Figura 3.13: Envio de *moveAck*.

Figura 3.14: Deslocamento do robô ao ponto de falha.

com o novo sensor já em operação, envia um *announce* ao *quorum*, que poderá ser diferente do anterior, dada sua movimentação para concluir o atendimento (Figura 3.15).

É importante salientar que a *periodicidade dos anúncios*, o *tempo de validade dos anúncios*, o *número de infoResponses* a ser esperado, o *tempo de espera por infoResponses*, o *tempo de espera por moveAck*, assim como os pesos utilizados para o cálculo do CA, devem ser escolhidos de modo a se adequar melhor à rede em que o SRRP vai ser utilizado.

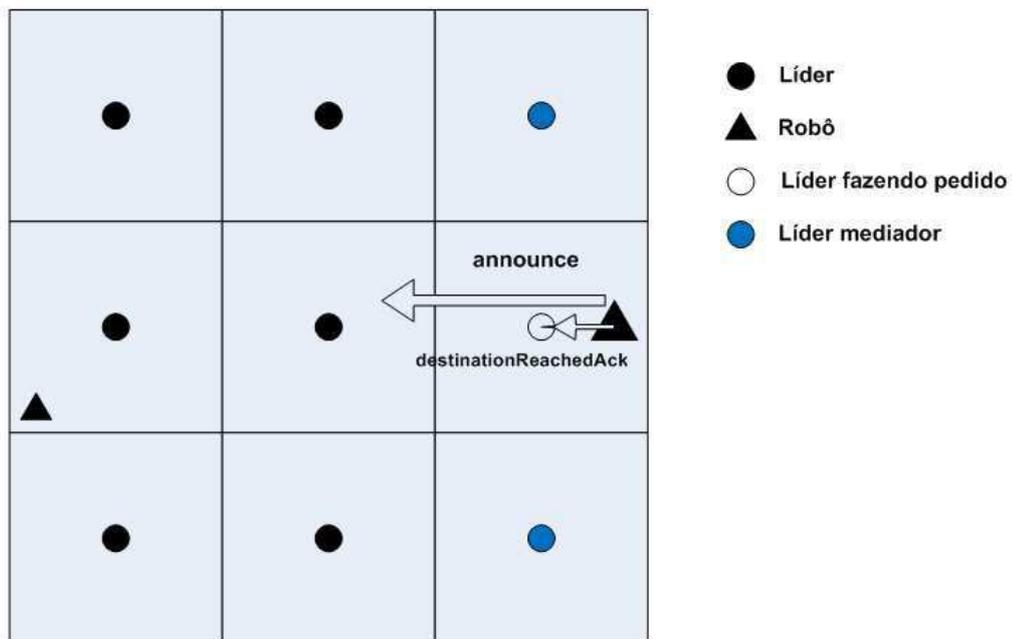


Figura 3.15: Envio de *destinationReachedAck* e *announce* em novo *quorum*.

Capítulo 4

Apresentação e Análise dos Resultados

Como metodologia seguida para conduzir a avaliação do desempenho do SRRP, foram realizadas simulações do protocolo em cenários pertinentes.

O simulador escolhido para a implementação do protocolo foi o NS-2 [15; 21], decisão esta pautada no fato do seu amplo uso do mesmo pela comunidade científica para experimentos dessa natureza, que se deve aos resultados significativos obtidos quando essa ferramenta é utilizada. A implementação do SRRP para o NS-2 está disponível em [2].

4.1 Ambiente das simulações

As simulações foram realizadas com a versão 2.34 do NS-2. Vários cenários de simulação foram elaborados, variando-se valores relevantes como o alcance do rádio de transmissão/recepção dos sensores e robôs, a velocidade dos robôs, o número de robôs existentes na rede e os valores que integram o coeficiente de atendimento – o CA, valores estes que chamados de fatores de decisão.

Como métricas para se medir o desempenho dos resultados das simulações foram utilizados o tempo médio de atendimento a uma requisição de ajuda e a porcentagem de tempo de rede coberta e conectada. O tempo médio de atendimento é contado a partir do momento exato em que um líder identifica um ponto de falha e inicia um pedido, até o momento em que recebe um *destinationReachedAck* de um robô, denotando que este chegou àquele ponto e atendeu ao pedido. A porcentagem de tempo de rede coberta e conectada, que será chamada apenas de tempo de rede coberta, é a porcentagem do tempo de simulação total em que

não havia nenhum pedido em andamento.

Foram utilizados 13 cenários, cujas descrições são apresentadas na Tabela 4.1. Para cada cenário foram executadas 100 rodadas de simulação, resultando em 1300 rodadas no total. A decisão do tamanho da amostra (100) se pautou em testes estatísticos para cálculo do tamanho da amostra com 95% de confiança. Cada rodada foi executada por 86400 segundos, um dia, em tempo do simulador, o que levava de 15 a 25 minutos em tempo real. Os pontos de falha foram pseudo-aleatoriamente gerados segundo uma distribuição exponencial (sem memória) com taxa de 1800 segundos; ou seja, sendo gerado, em média, um ponto de falha a cada 30 minutos. Foram escolhidas cem sementes diferentes para os geradores aleatórios e cada uma delas foi utilizada em uma das rodadas de um cenário, para que assim não houvesse geração das mesmas falhas em rodadas diferentes do mesmo cenário, mas as mesmas falhas pudessem ser repetidas para os diferentes cenários, o que leva a uma comparação mais fidedigna dos resultados.

Dado que a simulação é de um dia e que a taxa de chegada de pedidos é 30 minutos em média, esses dois fatores juntos fazem com que as simulações estressem significativamente o sistema.

Todos os valores foram representados com um intervalo de confiança de 95% para a média dos valores obtidos das cem rodadas de cada cenário.

Para todos os cenários foi utilizada uma área de 600m x 600m, dividida em 36 células de 100m x 100m. Em cada célula posicionou-se três sensores, um que desempenha o papel do líder, posicionado bem no meio da célula e outros dois formando com o líder uma diagonal na célula, totalizando 108 sensores (Figura 4.1).

Depois de decidida a grade que representa a rede, foram decididos também os valores configuráveis do protocolo:

- A *periodicidade dos anúncios* foi determinada como 100 segundos, pois estimou-se que seria um tempo razoável para um robô negociar um pedido e sair para atendê-lo. Se isso não ocorrer, o robô vai enviar outro announce esgotados os 100 segundos;
- O *tempo de validade dos anúncios* foi determinado como 300 segundos, pois assim, mesmo que o líder perca dois anúncios seguidos do mesmo robô, a validade ainda vai permitir que o robô esteja na lista do líder.

- O número de *infoResponses* foi determinado como 5. Já que nos cenários o número máximo de robôs é 18, 5 respostas perfazem 27,7% do total de respostas possíveis, valor considerável, já que não seria viável esperar por muitas respostas.
- O tempo de espera por *infoResponses* foi determinado como 120 segundos. Pensou-se primeiro no mesmo valor da *periodicidade dos anúncios*, mas foi dada uma tolerância de mais 20 segundos ao considerar situações em que alguns robôs pudessem estar atendendo a outras solicitações, o que atrasaria um pouco sua resposta.
- O tempo de espera por *moveAck* foi determinado como 60 segundos para que um líder não espere muito pela resposta de um robô, já que seu pedido deve ser atendido logo para recobrar a qualidade de serviço da rede.

Na Tabela 4.2 estão sumarizadas as configurações adotadas.

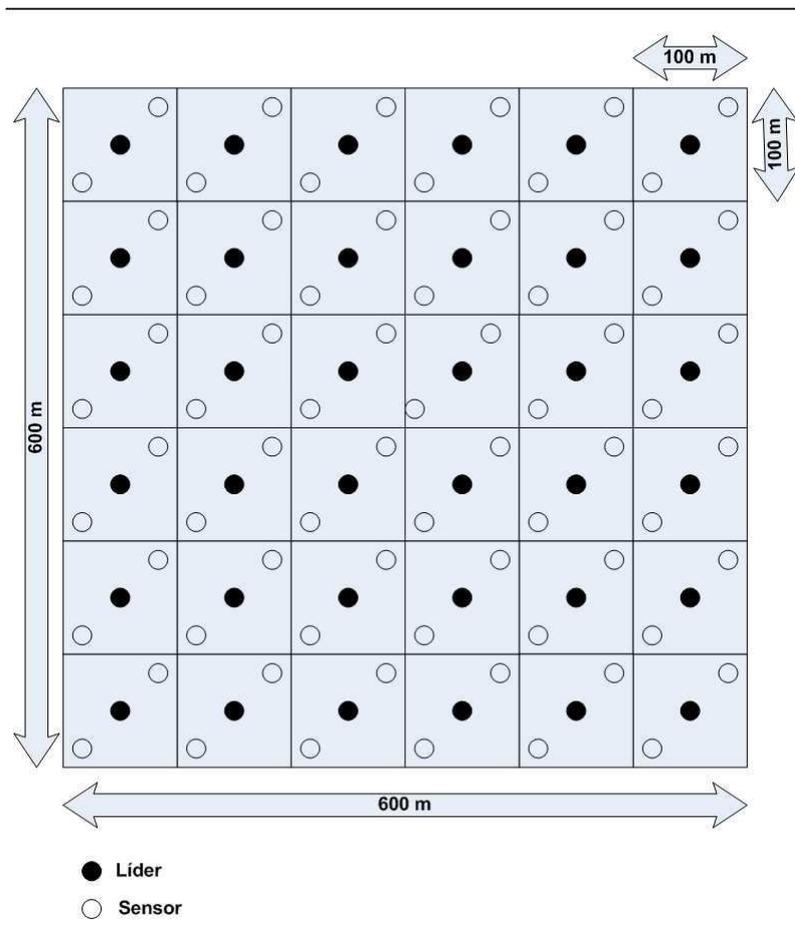


Figura 4.1: Grade representando a área da rede (ainda sem robôs)

| Cenário | Alcance do rádio (m) | Velocidade dos robôs (m/s) | Número de robôs | Peso dos fatores de decisão |
|---------|----------------------|----------------------------|-----------------|---|
| 1 | 50 | 1.5 | 9 | 33% para cada fator |
| 2 | 75 | 1.5 | 9 | 33% para cada fator |
| 3 | 100 | 1.5 | 9 | 33% para cada fator |
| 4 | 50 | 2.5 | 9 | 33% para cada fator |
| 5 | 50 | 4.0 | 9 | 33% para cada fator |
| 6 | 75 | 2.5 | 9 | 33% para cada fator |
| 7 | 75 | 4.0 | 9 | 33% para cada fator |
| 8 | 75 | 1.5 | 12 | 33% para cada fator |
| 9 | 75 | 1.5 | 18 | 33% para cada fator |
| 10 | 75 | 1.5 | 9 | 50% para distância, 25% para energia, 25% para n° de sensores |
| 11 | 75 | 1.5 | 9 | 50% para energia, 25% para distância, 25% para n° de sensores |
| 12 | 75 | 1.5 | 9 | 50% para n° de sensores, 25% para energia, 25% para distância |
| 13 | 75 | 2.5 | 12 | 50% para distância, 25% para energia, 25% para n° de sensores |

Tabela 4.1: Descrição dos cenários.

| | |
|--|------------------------|
| Simulador | Network Simulator 2.34 |
| Rodadas por cenário | 100 |
| Duração das rodadas | 86400 segundos |
| Intervalo entre a geração de pontos de falha | 30 minutos, em média |
| Área da simulação | 600m x 600m |
| Células | 36 de 100m x 100m cada |
| Sensores | 108, sendo 36 líderes |
| Valores Configuráveis do Protocolo | |
| Periodicidade dos anúncios | 100s |
| Tempo de validade dos anúncios | 300s |
| Número de <i>infoResponses</i> | 5 |
| Tempo de espera por <i>infoResponses</i> | 120s |
| Tempo de espera por <i>moveAck</i> | 60s |

Tabela 4.2: Ambiente de simulações do SRRP.

4.2 Experimentos

4.2.1 Variando o alcance do rádio de transmissão/recepção dos nós

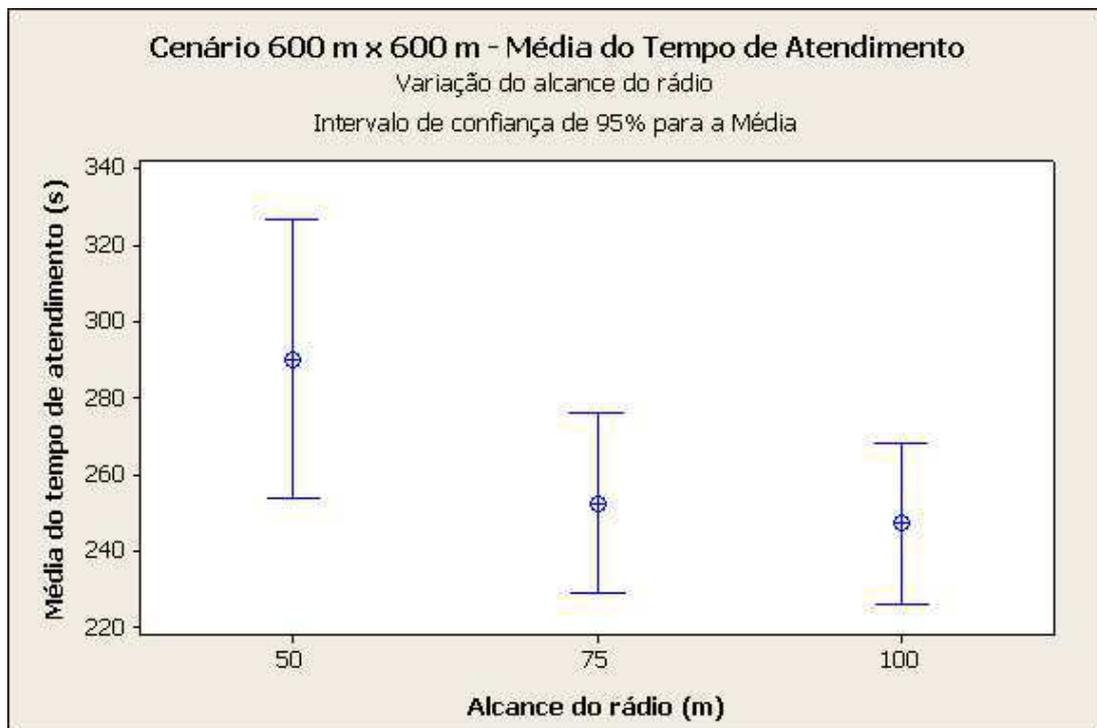
O objetivo desse experimento é observar o impacto do uso de diferentes valores de alcance do rádio de transmissão/recepção dos nós (sensores, líderes e robôs) no desempenho do protocolo, almejando conhecer um valor adequado que conseguisse um desempenho satisfatório sem consumir muita energia das baterias dos nós, já que o rádio é grande vilão nesse sentido [22].

Foram utilizados três cenários diferentes para o alcance de rádio, com valores de 50 m, 75 m e 100 m. Os valores dos resultados das simulações podem ser conferidos na Tabela 4.3 e nos gráficos apresentados pelas Figura 4.2 e Figura 4.3.

Dada a configuração diagonal da posição dos sensores em cada célula, sendo estas de 100m x 100m, o alcance do rádio deveria ser de pelo menos 70.71m para que a informação pudesse fluir pelos sensores da célula. Usando alcance de 50m, a informação só consegue fluir por causa do posicionamento dos robôs (os robôs usam o mesmo alcance de rádio),

| Alcance do rádio (m) | Média de tempo de atendimento (s) | | Tempo de rede coberta (%) | |
|----------------------|-----------------------------------|------------------------|---------------------------|------------------------|
| | Média | Intervalo de confiança | Média | Intervalo de confiança |
| 50 | 290.1 | (253.5; 326.8) | 84.13 | (82.03; 86.23) |
| 75 | 252.4 | (228.85; 272.86) | 85.6 | (83.61; 87.61) |
| 100 | 247.2 | (226.07; 268.25) | 86.4 | (85.1; 87.77) |

Tabela 4.3: Variação do alcance do rádio.

Figura 4.2: Gráfico do alcance do rádio *versus* a média do tempo de atendimento.

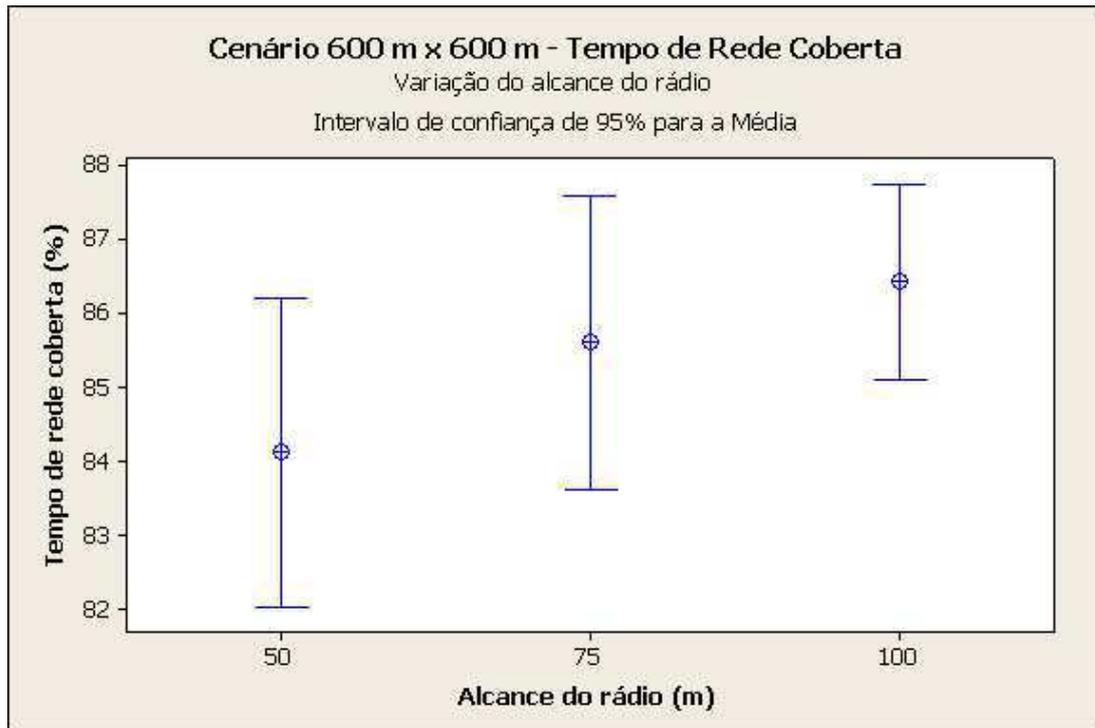


Figura 4.3: Gráfico do alcance do rádio *versus* o tempo de rede coberta.

que acaba por interligar alguns dos sensores, pelos quais circula o fluxo de mensagens. Este fato acaba por atrasar o fluxo de informações, fazendo-o percorrer maior distância, atrasando também todo o processo do protocolo. Por isso, a média de tempo de atendimento é 15% maior que a obtida com o rádio de 75m; e o tempo de rede coberta é 1.5% menor.

A grande diferença entre as porcentagens de ganho no atendimento e no tempo de rede coberta desses cenários (de 15% para 1.5%) e também de alguns outros cenários apresentados no decorrer do capítulo, deve-se ao fato de que os cenários com melhores desempenhos no tempo de atendimento conseguem atender a mais pedidos durante o tempo de simulação, já que os líderes só tratam uma falha por vez. Aumentando o número de pedidos, inevitavelmente se aumenta o tempo em que há pedidos sendo processados.

Entre os rádios com alcance de 75m e 100 m, a diferença não é muito significativa, sendo a média de tempo de atendimento usando o rádio de 75m 2.1% maior e o tempo de rede coberta, 0.8% menor. Isso se deve ao fato do uso do *broadcast* controlado nos pedidos e anúncios, fazendo com que o alcance do rádio de 100m seja eficiente em apenas uma direção (vertical ou horizontal, dependendo do tipo do pacote) e acarrete vários descartes de pacotes na outra direção.

Com base nesses resultados, pode-se perceber que usar um rádio com alcance de 75m seria, dentre as opções testadas, a com melhor custo/benefício, pois obteve resultados bem melhores que o rádio com 50m de alcance e um ganho apenas pouco menor que o rádio com 100m, consumindo, porém, menos energia dos nós. Entretanto, em cenários nos quais qualquer ganho no tempo atendimento ou no tempo de rede coberta é importante, a escolha mais apropriada seria a do rádio de 100m de alcance.

4.2.2 Variando a velocidade dos robôs

O presente experimento tem a pretensão de delinear o quanto a velocidade do robô pode impactar no desempenho do protocolo para, assim, poder escolher robôs com velocidade que possa melhorar o desempenho do protocolo, mas que não sejam muito caros, já que um robô com maior velocidade poderia ter maior custo de produção em alguns casos.

Foram feitas rodadas em três cenários utilizando diferentes velocidades para os robôs, 1.5 m/s, 2.5 m/s, 4.0 m/s e alcance de rádio de 50m; e outras três utilizando, agora, alcance de rádio de 75m. Os resultados e análises de cada grupo de cenários serão apresentados a seguir.

Variação da velocidade dos robôs utilizando alcance de rádio de 50m

Os resultados obtidos para esse grupo de cenários podem ser conferidos na Tabela 4.4 e nos gráficos apresentados pelas Figura 4.4 e Figura 4.5.

| Velocidade dos robôs (m/s) | Média de tempo de atendimento (s) | | Tempo de rede coberta (%) | |
|----------------------------|-----------------------------------|------------------------|---------------------------|------------------------|
| | Média | Intervalo de confiança | Média | Intervalo de confiança |
| 1.5 | 290.1 | (253.5; 326.8) | 84.13 | (82.03; 86.23) |
| 2.5 | 196.8 | (171.85; 221.75) | 89.35 | (87.95; 90.74) |
| 4.0 | 179.0 | (152.8; 205.2) | 90.12 | (88.53; 91.51) |

Tabela 4.4: Variação da velocidade dos robôs utilizando rádio de 50m.

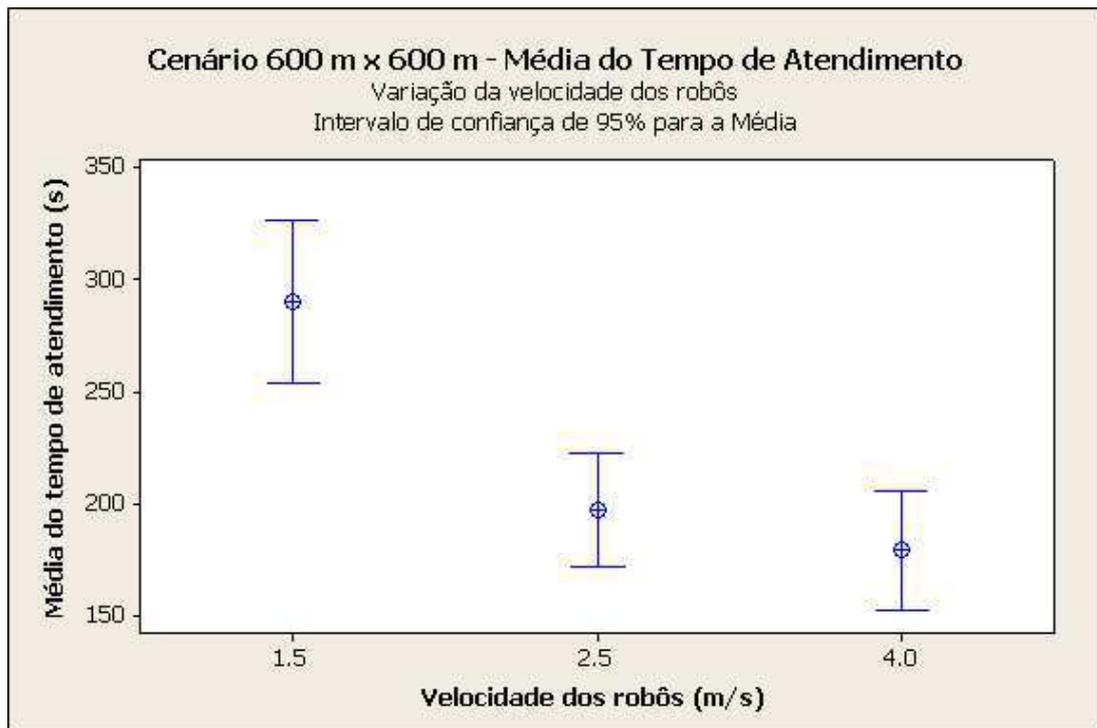


Figura 4.4: Gráfico da velocidade dos robôs *versus* a média do tempo de atendimento (rádio de 50m).

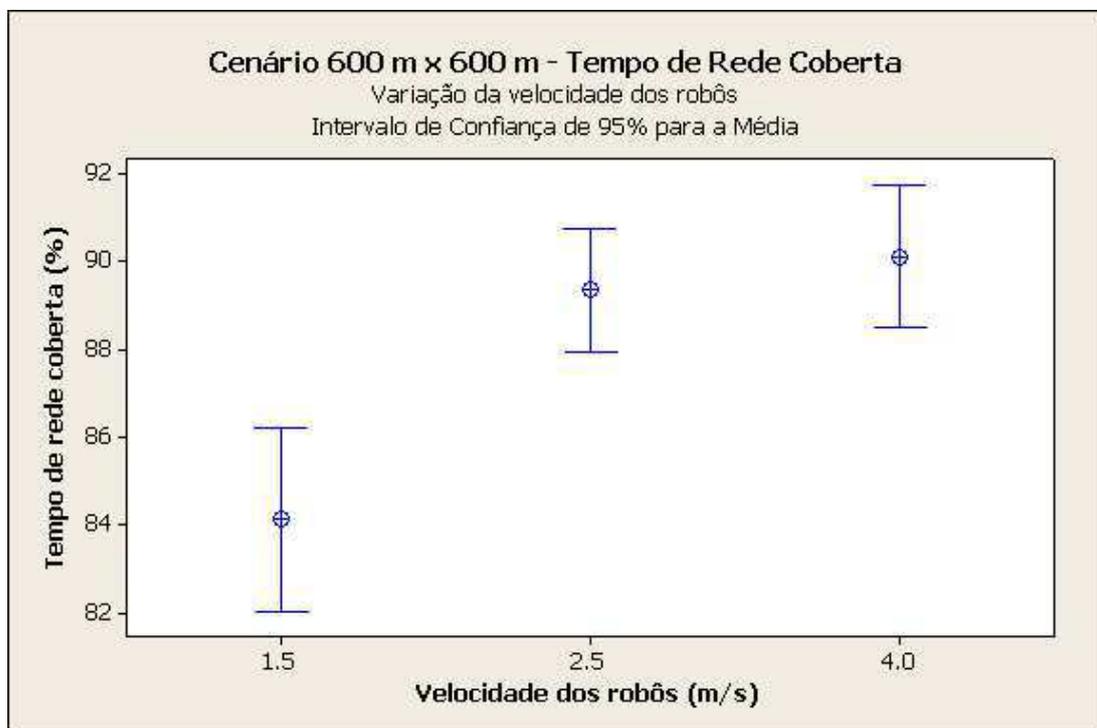


Figura 4.5: Gráfico da velocidade dos robôs *versus* o tempo de rede coberta (rádio de 50m).

Percebe-se que houve uma diminuição significativa (32.17%) na média do tempo de atendimento e um aumento de 5% do tempo total de rede coberta usando robôs com velocidade de 2.5 m/s em relação ao uso de robôs com 1.5 m/s, ganhos que se devem, obviamente, ao fato de os robôs atenderem mais rapidamente aos pedidos, já que possuem maior velocidade.

Já usando robôs com velocidade de 4.0 m/s, o ganho na média de tempo de atendimento em relação a quando se usa robôs com 1.5 m/s foi de 38.3% e no tempo total de rede coberta, foi de 6%.

A melhoria no desempenho usando robôs com 4.0 m/s de velocidade não cresce à mesma taxa que usando velocidade de 2.5 m/s, como seria o esperado, pois os robôs gastam muito mais energia para poder atingir essa velocidade. Verificando os arquivos de *trace*, nota-se que os primeiros atendimentos são bem mais rápidos, porém, à medida que os robôs vão muito mais rapidamente acabando com sua energia, como esta é uma métrica de decisão sobre qual robô vai atender a um pedido, um robô mais próximo pode não ser escolhido por sua energia estar quase esgotada, levando a atendimentos mais demorados dada a escolha de robôs mais distantes. Também, alguns robôs mais solicitados ficam totalmente sem energia muito mais cedo que nos outros casos, diminuindo assim a quantidade de robôs na rede, o que leva a sobrecarregar os outros. Mesmo assim, o ganho ainda é um pouco maior.

De acordo com os resultados apresentados, a melhor velocidade a ser escolhida, caso o custo de produção dos robôs com maior velocidade fosse muito maior e estourasse o orçamento, seria a de 2.5 m/s, já que a diferença de ganho quando há o uso da velocidade de 4.0 m/s não foi tão grande e seriam robôs mais baratos a serem produzidos. Entretanto, se a velocidade de atendimento a um pedido for tida como elemento essencial, a velocidade de 4.0 m/s deveria ser a escolhida.

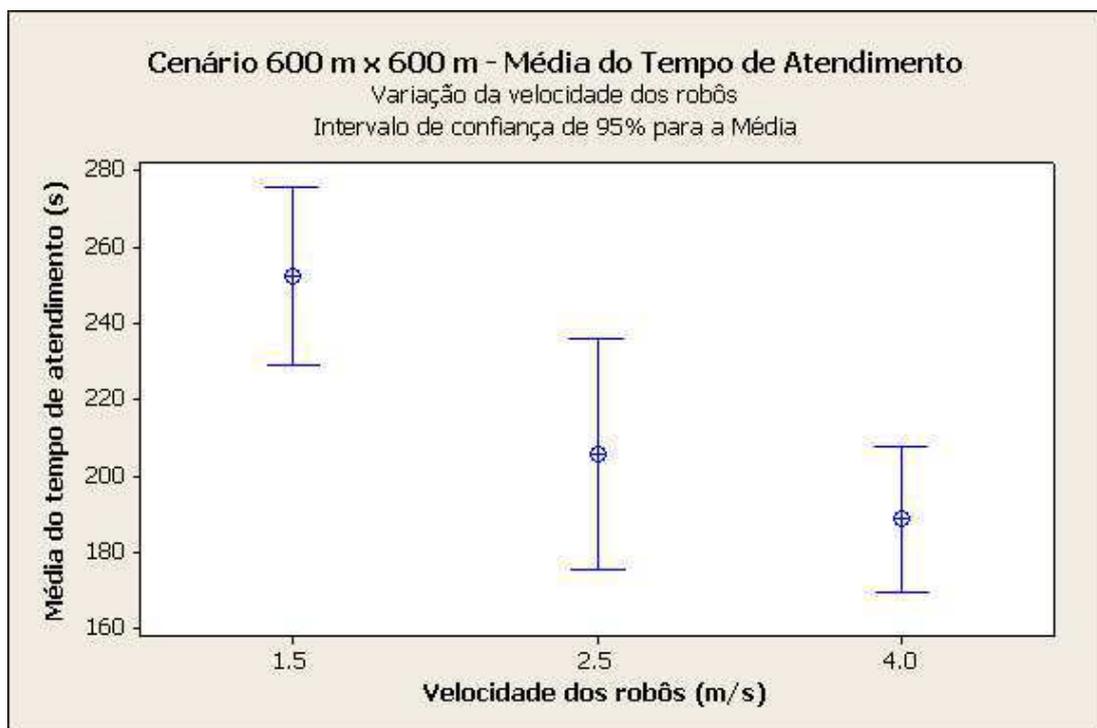
Variação da velocidade dos robôs utilizando alcance de rádio de 75m

Os resultados obtidos com a variação da velocidade dos robôs utilizando nós com rádio de alcance de 75m podem ser visualizados na Tabela 4.5 e nas Figura 4.6 e Figura 4.7 que apresentam, respectivamente, o gráfico da média de tempo de atendimento e da porcentagem de tempo de rede coberta.

Pode-se perceber que houve uma melhoria no desempenho em relação ao experimento anterior, em que foi usado um alcance de rádio de 50m. Como exemplo, houve uma redução

| Velocidade dos robôs (m/s) | Média de tempo de atendimento (s) | | Tempo de rede coberta (%) | |
|----------------------------|-----------------------------------|------------------------|---------------------------|------------------------|
| | Média | Intervalo de confiança | Média | Intervalo de confiança |
| 1.5 | 252.4 | (228.85; 272.86) | 85.6 | (83.61; 87.61) |
| 2.5 | 205.5 | (175.25; 235.75) | 88.82 | (87.01; 90.62) |
| 4.0 | 188.55 | (169.4; 207.7) | 89.71 | (88.5; 90.92) |

Tabela 4.5: Variação da velocidade dos robôs utilizando rádio de 75m.

Figura 4.6: Gráfico da velocidade dos robôs *versus* a média do tempo de atendimento (rádio de 75m).

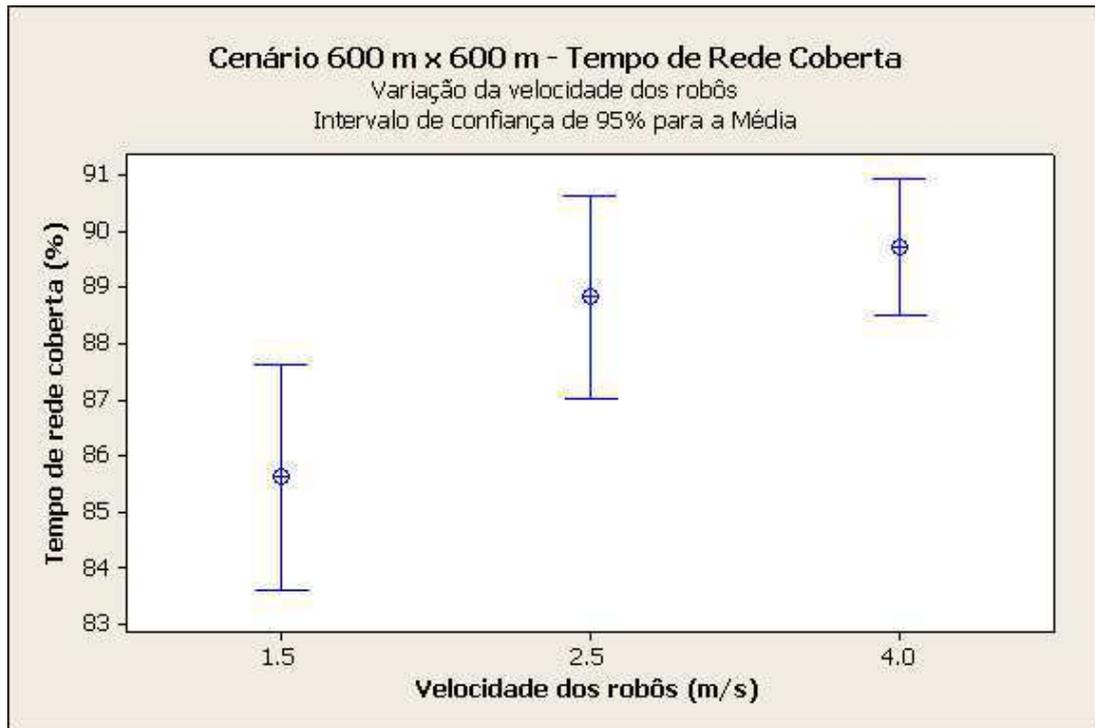


Figura 4.7: Gráfico da velocidade dos robôs *versus* o tempo de rede coberta (rádio de 75m).

de 13% na média do tempo de atendimento e um aumento de 1.48% no tempo de rede coberta em se comparando os dois cenários que usam velocidade de 1.5 m/s. Isso corrobora o resultado do experimento explicitado na Seção 4.2.1, mostrando que com um alcance de rádio de 75m, o protocolo funciona de maneira mais eficaz, dado o posicionamento dos sensores.

Comparando-se agora a variação de velocidades desse experimento, temos uma redução de 18.6% na média do tempo de atendimento e aumento de 3.2% no tempo de rede coberta, quando se utiliza velocidade de 2.5% em relação a de 1.5%. E redução de 25.3% na média do tempo de atendimento e aumento de 4.1% no tempo de rede coberta, quando se utiliza velocidade de 4.0 m/s em relação a de 1.5 m/s.

Apesar dos ganhos terem se dado em menor escala que os do experimento anterior, e isso se deve ao uso do rádio de 75m que acaba por uniformizar a comunicação, não deixando o fluxo de informações seguindo apenas por intermédio dos robôs, o comportamento foi bem parecido, tendo ganhos do uso da velocidade de 4.0 m/s sobre a de 2.5 m/s e esta, sobre a de 1.5 m/s, não se dando, porém, esses ganhos a uma mesma taxa, pelos mesmos motivos já apresentados na Seção 4.2.2.1.

4.2.3 Variando o número de robôs

Variando agora o número de robôs que são implantados na rede, almeja-se medir que importância esse fator tem em relação ao desempenho do protocolo. Até então, estavam sendo utilizados 9 robôs, ou seja, 25% do número de células da rede, como se cada um tivesse, em média, que atender aos pedidos de 4 células. Foram feitas também simulações utilizando 12 robôs (33.33% do número de células) e 18 robôs (50% do número de células).

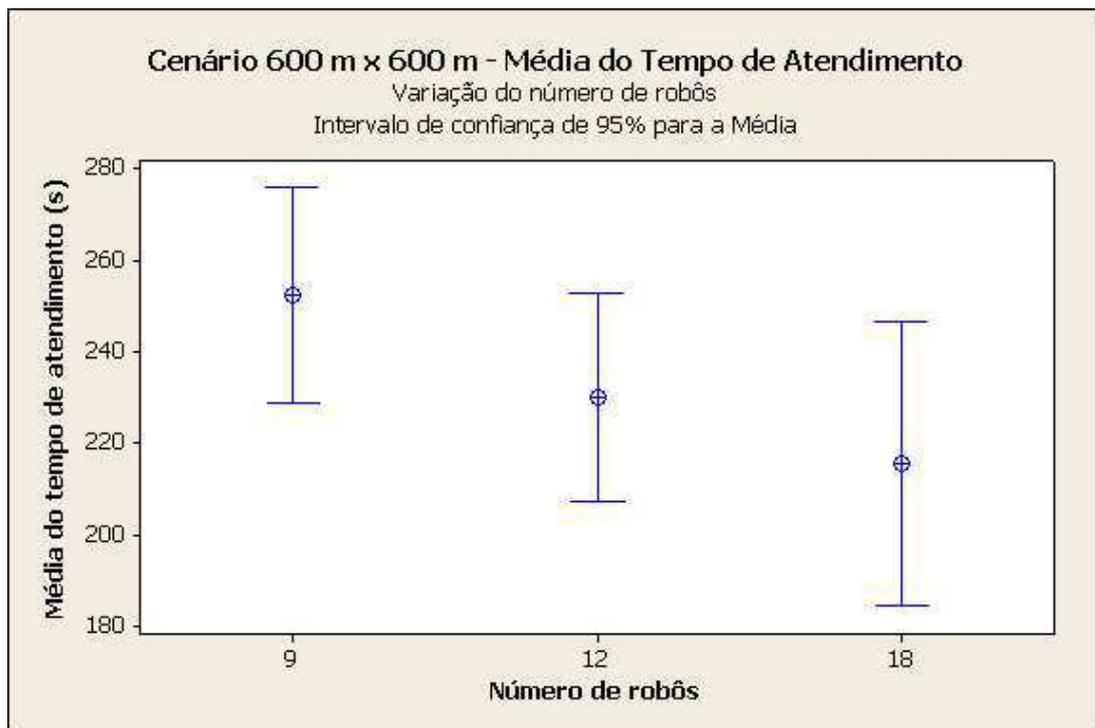
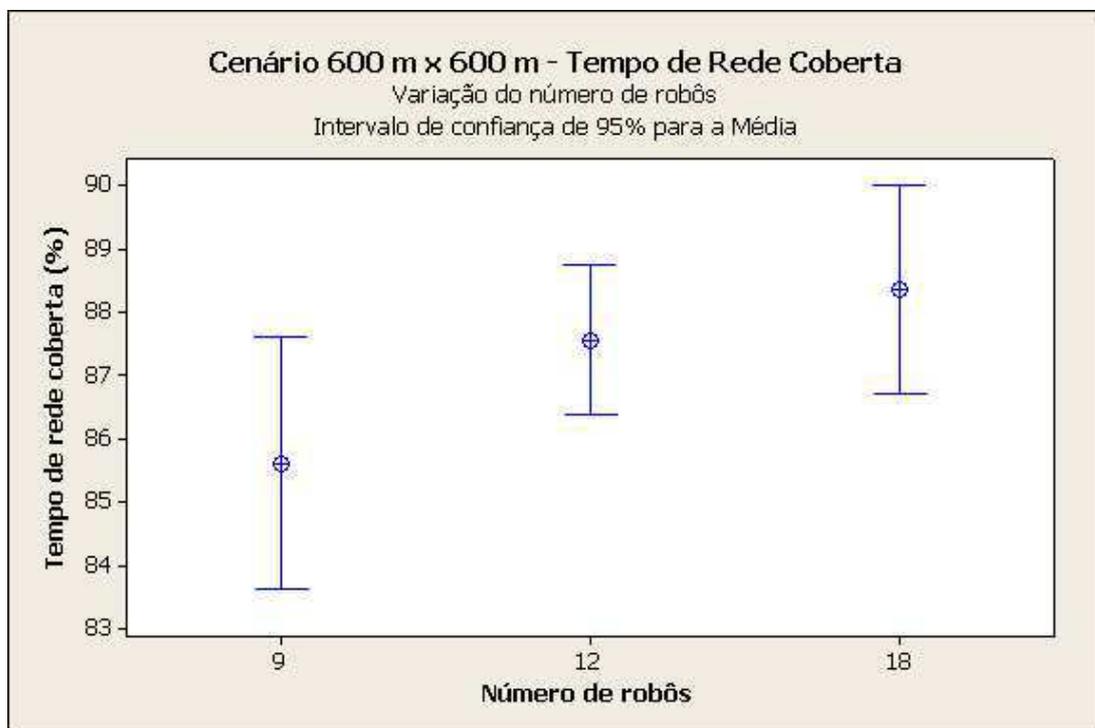
Nesses cenários foram utilizados o alcance de rádio de 75m e 1.5 m/s como velocidade dos robôs. Os robôs foram uniformemente distribuídos na área da rede. Os resultados são apresentados na Tabela 4.6 e nos gráficos das Figura 4.8 e Figura 4.9.

| Número de robôs | Média de tempo de atendimento (s) | | Tempo de rede coberta (%) | |
|-----------------|-----------------------------------|------------------------|---------------------------|------------------------|
| | Média | Intervalo de confiança | Média | Intervalo de confiança |
| 9 | 252.4 | (228.85; 272.86) | 85.6 | (83.61; 87.61) |
| 12 | 229.9 | (207.23; 252.64) | 87.57 | (86.38; 88.75) |
| 18 | 215.4 | (184.42; 246.39) | 88.37 | (86.71; 90.03) |

Tabela 4.6: Variação do número de robôs.

Observa-se nos resultados que o cenário com 12 robôs obteve melhor desempenho sobre aquele com 9 robôs, tendo uma redução de 8.92% do tempo médio de atendimento e um aumento de 1.95% no tempo de rede coberta. Já o cenário de 18 robôs obteve uma redução de 14.65% do tempo médio de atendimento e um aumento de 2.75% no tempo de rede coberta em relação àquele com 9 robôs.

Como era de se esperar, há ganho em se aumentar o número de robôs na rede, não havendo gargalo de desempenho pelo aumento do número de mensagens que nela transitam. Também aqui, o ganho em usar 18 robôs sobre usar 9 deles não se deu à mesma taxa que usando 12 robôs (apesar de já ter se aproximado mais em relação aos experimentos de velocidade), devido ao fato de que o mesmo número de falhas foi gerado e apenas os robôs se dividiram no atendimento às mesmas, não tendo sido gerado um maior número falhas para o cenário com 18 robôs. Com base em análise dos arquivos de *trace*, percebeu-se que no caso do uso de 18 robôs, um maior número deles ficava ocioso e por mais tempo, esperando novo

Figura 4.8: Gráfico do número de robôs *versus* a média do tempo de atendimento.Figura 4.9: Gráfico do número de robôs *versus* o tempo de rede coberta.

pedido. Porém, eles também terminavam as rodadas com mais energia remanescente, já que, em geral, trabalhavam menos. Isso seria bem importante para se ponderar ao aplicar o SRRP em redes cujas missões são muito longas e com grande risco, ou seja, alta probabilidade de falhas, visando assim a maior durabilidade do nível de energia dos robôs e, consequentemente, um maior tempo de validade de seus serviços para a rede, o que acarreta um maior tempo de vida para a mesma.

4.2.4 Variando os pesos dos fatores de decisão do coeficiente de atendimento

Neste experimento, foram variados os pesos dos fatores de decisão do coeficiente de atendimento, que são a distância do robô ao ponto de falha, a quantidade de energia remanescente na bateria do robô e o número de sensores que o robô ainda carrega consigo. O CA, como já explicitado anteriormente, é utilizado pelo líder que está fazendo um pedido para comparar os robôs e escolher aquele que melhor pode lhe atender.

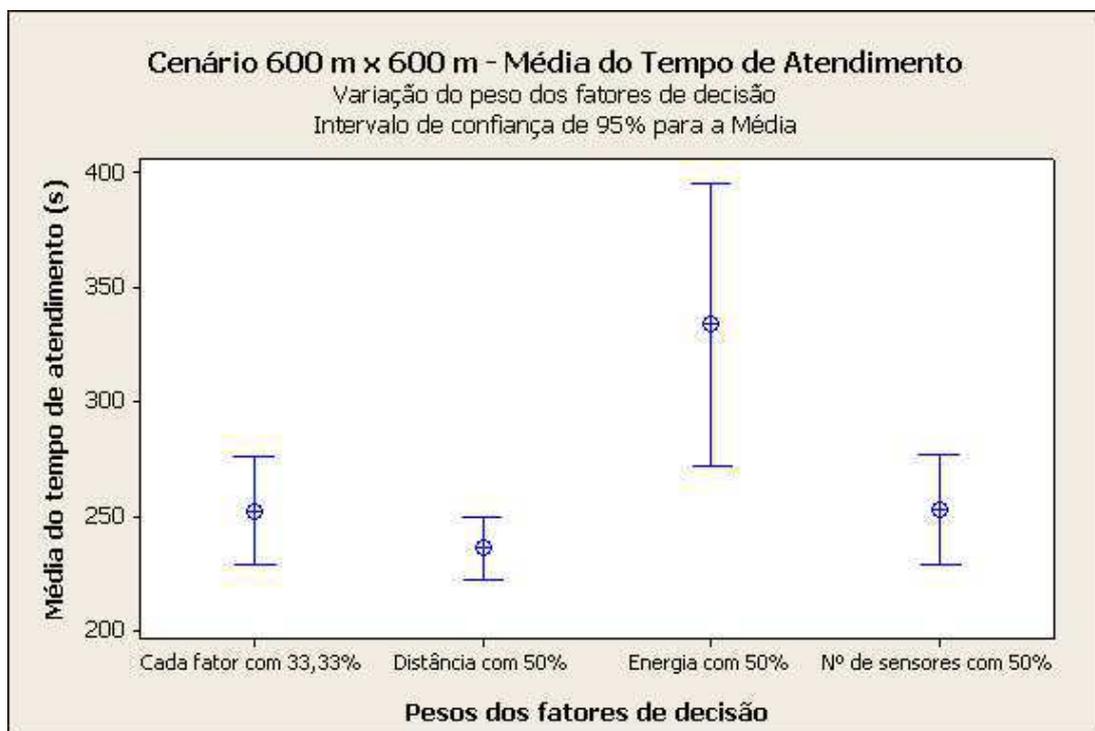
Até então, os experimentos tinham sido executados com cada um desses fatores possuindo um peso igual (33.33%), ou seja, nenhum fator era mais decisivo que outro para compor o valor do CA do robô. Além desse cenário, foram realizadas simulações com outros três: um com a distância tendo peso de 50% e os demais fatores com 25% cada; outro com a quantidade de energia tendo peso de 50% e os demais fatores com 25% cada; e um último, com o número de sensores do robô tendo peso de 50% e os demais fatores com 25% cada.

As rodadas foram realizadas utilizando rádio com 75m de alcance, robôs com 1.5 m/s de velocidade e 9 robôs distribuídos pela rede. Os resultados podem ser conhecidos na Tabela 4.7 e nos gráficos apresentados na Figura 4.10 e na Figura 4.11.

Os resultados mostram que o protocolo teve melhor desempenho quando a distância entre o robô e o ponto de falha teve maior peso. Esse cenário teve uma redução de 6.5% na média de tempo de atendimento e um aumento de 1.36% no tempo de rede coberta, em relação ao cenário em que os pesos eram iguais para todos os fatores. Isto se deve ao fato de que, se todos os robôs possuem a mesma velocidade, e um robô está mais perto do ponto de falha, ele vai chegar mais rápido para atender ao pedido, a menos que acabe a sua energia, porém,

| Pesos dos fatores de decisão | Média de tempo de atendimento (s) | | Tempo de rede coberta (%) | |
|------------------------------|-----------------------------------|------------------------|---------------------------|------------------------|
| | Média | Intervalo de confiança | Média | Intervalo de confiança |
| Cada fator com 33.33% | 252.4 | (228.85; 272.86) | 85.6 | (83.61; 87.61) |
| Distância com 50% | 235.91 | (221.94; 249.88) | 87.27 | (86.47; 88.06) |
| Energia com 50% | 333.8 | (272.18; 395.49) | 81.42 | (77.7; 85.15) |
| Número de sensores com 50% | 252.9 | (228.97; 276.87) | 86.38 | (85.05; 87.72) |

Tabela 4.7: Variação dos pesos dos fatores de decisão.

Figura 4.10: Gráfico dos pesos dos fatores de decisão *versus* a média do tempo de atendimento.

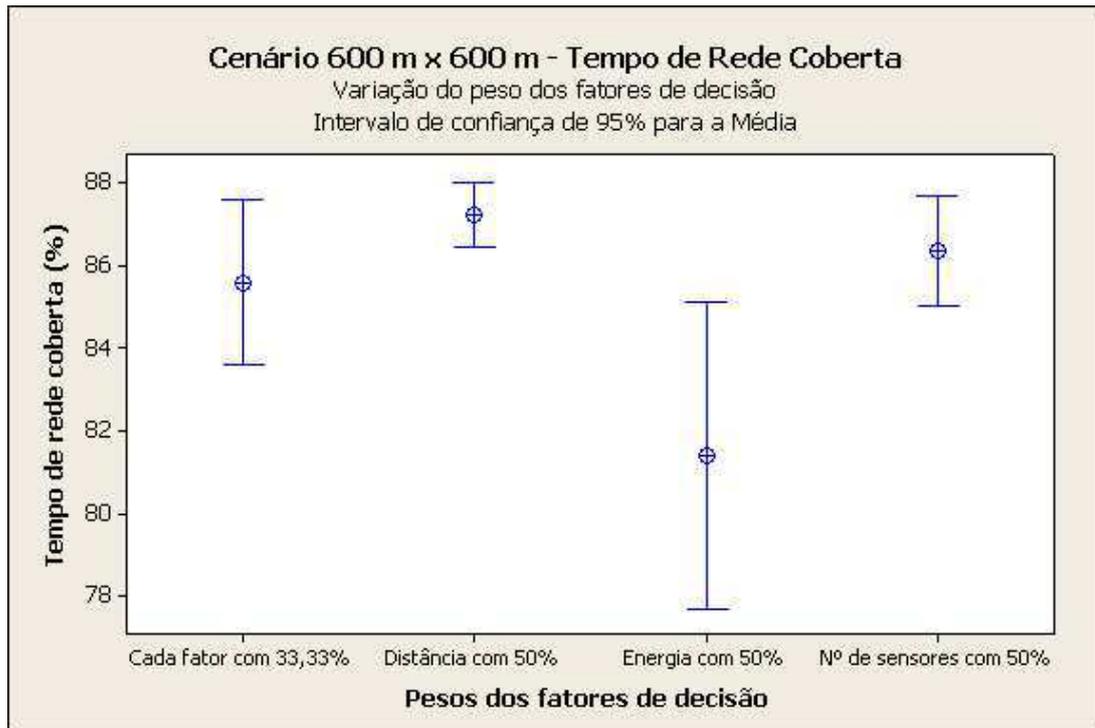


Figura 4.11: Gráfico dos pesos dos fatores de decisão *versus* o tempo de rede coberta.

a energia também tem um peso na decisão, o que leva a este evento só ocorrer nos casos em que também um robô mais distanciado não conseguiria se movimentar até a falha. Além do mais, o robô mais próximo também consumirá menos energia para atingir o ponto de falha.

Pode-se perceber também que os resultados das simulações do cenário em que o número de sensores possui peso de 50% não diferenciam significativamente do cenário com pesos iguais para todos, tendo apenas um ínfima queda de desempenho, com aumento de 0.2% na média do tempo de atendimento e uma redução de 0.78% no tempo de rede coberta. Analisando os arquivos de *trace*, percebeu-se que os pedidos estavam sendo bem distribuídos entre os robôs, levando-os a utilizar quase o mesmo número de sensores. Como essa métrica era praticamente igual, a formação do CA e a consequente escolha do robô ficava a cargo dos outros dois fatores, distância e energia. Como estes tinham mesmo peso, assim como no cenário de pesos iguais para todos os fatores, as decisões foram, com mínima diferença, as mesmas.

Já o desempenho obtido com a energia sendo o principal fator decisório (com peso de 50%) foi inferior ao do cenário com pesos iguais, tendo apresentado um aumento de 32.25% na média do tempo de atendimento e uma redução de 4.2% no tempo de rede coberta. Isto

se deve ao fato de robôs muito distantes terem sido escolhidos para atender a uma falha por ainda não terem trabalhado e, assim, possuírem mais energia que um outro robô que estava bem próximo à falha. Só que por estarem distantes, os robôs, acabavam gastando muita energia para atender àquele pedido, levando-os a uma exaustão de energia bem mais precoce que nos outros cenários, o que acabava por deixar vários pedidos em aberto e outros sem atendimento. Além do mais, por estarem distantes e com a mesma velocidade, os robôs demoravam muito mais para conseguir chegar ao ponto de falha.

4.2.5 Experimento otimizado

Diante dos resultados obtidos, decidiu-se realizar a simulação de um cenário otimizado, utilizando parâmetros que apresentaram melhores resultados, considerando a relação custo/-benefício. O cenário otimizado utiliza a seguinte configuração:

- Rádio com alcance de 75m: melhor desempenho que o de 50m e um desempenho próximo ao de 100m, mas que consumiria menos energia, além de ser mais barato em termos reais de custo financeiro;
- Velocidade dos robôs de 2.5 m/s: melhor relação custo/benefício relativo a custo de produção e de consumo de energia e desempenho;
- Número de robôs utilizados igual a 12: minimização custos (menos robôs na rede em relação ao de 18 robôs) sem que o desempenho seja reduzido;
- Coeficiente de atendimento com peso de 50% para a distância e 25% para cada um dos demais fatores: escolhido por ter apresentado o melhor desempenho, sem necessitar de aumento em custo com a estrutura da rede.

Este cenário foi comparado com um outro, chamado aqui de cenário base, que possui as características base de todos os experimentos, excetuando-se o alcance do rádio, que era de 50m. Ou seja, alcance de rádio de 75m, velocidade dos robôs de 1.5 m/s, 9 robôs, e fatores de decisão com pesos iguais de 33.33%.

Os resultados desse experimento podem ser conferidos na Tabela 4.8 e nos gráficos apresentados pelas Figura 4.12 e Figura 4.13.

| Cenário | Média de tempo de atendimento (s) | | Tempo de rede coberta (%) | |
|-----------|-----------------------------------|------------------------|---------------------------|------------------------|
| | Média | Intervalo de confiança | Média | Intervalo de confiança |
| Base | 252.4 | (228.85; 272.86) | 85.6 | (83.61; 87.61) |
| Otimizado | 164.7 | (156.46; 172.93) | 90.98 | (90.46; 91.5) |

Tabela 4.8: Resultados do cenário otimizado.

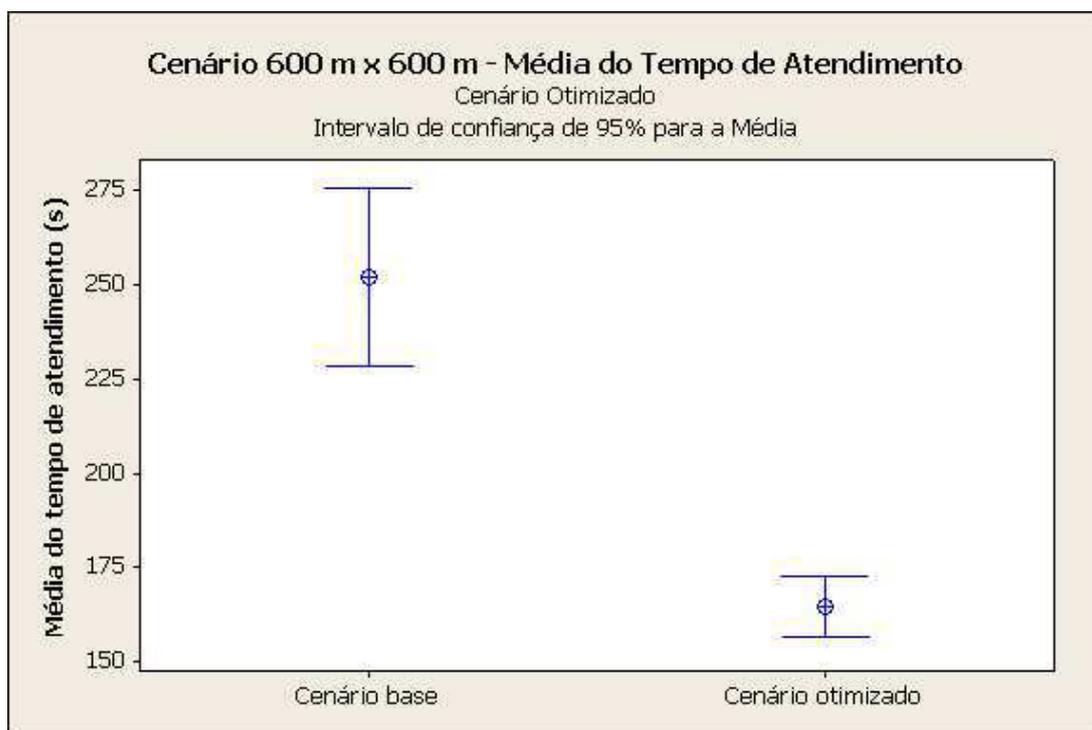


Figura 4.12: Gráfico do cenário otimizado: média do tempo de atendimento.

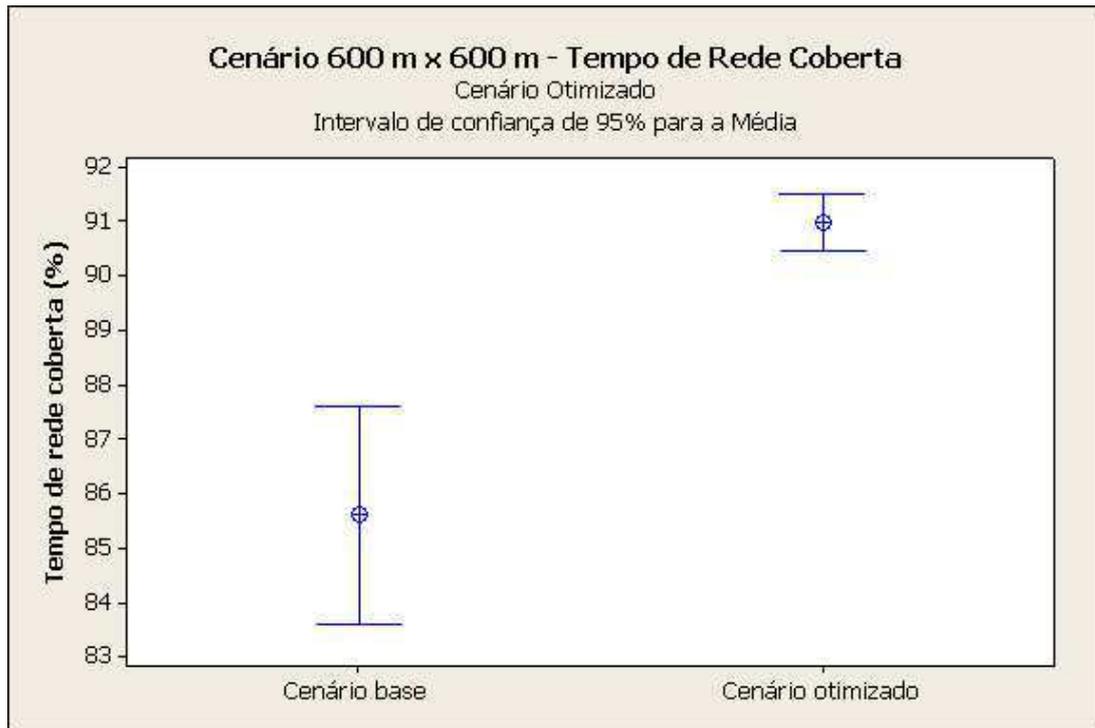


Figura 4.13: Gráfico do cenário otimizado: tempo de rede coberta.

Os resultados mostram que o ganho do cenário otimizado foi substancial, tendo diminuído em 34,75% a média do tempo de atendimento e aumentado 5,37% o tempo de rede coberta. Lembrando que o aumento deste último resultado não é tão maior por causa da taxa de geração de pedidos ser alta, o que estressa o sistema. Em situações mais amenas, esse ganho seria bem maior.

Sendo assim, chega-se à conclusão de que o cenário otimizado pode ser utilizado para obter um melhor desempenho sem que se aumente muito o custo com estrutura da rede em relação ao cenário base.

Capítulo 5

Considerações Finais

Neste capítulo são relacionadas as considerações finais do trabalho, incluindo as conclusões obtidas a partir dos resultados dos experimentos, as principais dificuldades enfrentadas e sugestões para trabalhos futuros.

Nesta dissertação foi apresentada uma solução de *redployment* para redes de sensores sem fio, visando recobrar um nível adequado de qualidade de serviço que vai sendo perdido durante a execução da rede devido a falha de nós. Essa solução utiliza robôs dotados de mobilidade, que carregam consigo sensores e os depositam, segundo demanda, em pontos onde houve falha de um ou mais sensores e que acarretaram em problemas na cobertura e conectividade da rede. Para isso, sensores tidos como líderes, sendo responsáveis por detectar essas falhas em uma parte da rede, chamam robôs a repor nós onde for preciso. A implementação do protocolo não compreende a detecção de pontos de falha, podendo ser utilizada uma das soluções já propostas por Wang *et al.* [33], Li *et al.* [23], Wu *et al.* [34] e Sheu *et al.* [29]. A rede é dividida em uma grade, havendo um líder responsável em cada célula.

Para que não haja grande tráfego de pacotes pela rede, o que, em redes de sensores sem fio, é primordial, já que os nós normalmente são providos de pouca energia e devem tentar economizá-la ao máximo, foi utilizada a idéia de *quorum*, em que as informações só fluem em parte da rede. Foram adotados dois tipos de *quorum*, o de anúncios e o de pedidos. Os robôs anunciam que no momento pertencem a determinado *quorum*, enviando mensagens para o quorum de anúncios, formado pelas células da linha em que o robô se encontra. Já os líderes fazem pedidos de ajuda enviando mensagem para seu *quorum* de pedidos, formados

pelas células da mesma coluna que o líder. Um líder que consegue interceptar os dois tipos de mensagem, responde ao líder do pedido com uma mensagem contendo o IP do robô, para que o líder possa entrar em contato com este. Utilizando esse esquema, as mensagens que seriam enviadas para toda a rede (anúncios e pedidos), apenas o serão para um grupo de células, diminuindo em muito o tráfego.

O SRRP é um protocolo que visa a manutenção da qualidade de serviço da rede sem a necessidade de acesso físico humano ao ambiente em que a rede está implantada, característica importante, já que o ambiente pode ser hostil e/ou de difícil acesso. Além disso, foi implementado o mecanismo do *broadcast* controlado, para que seja diminuído o consumo de energia com roteamento de pacotes pela rede.

Foram realizados experimentos que mediram o impacto do aumento do alcance do rádio dos sensores e robôs, do aumento da velocidade dos robôs, do aumento do número de robôs inseridos na rede e da variação dos pesos dos fatores de decisão na eficiência do protocolo.

Os experimentos foram importantes para mostrar, de acordo com a rede e inferências utilizadas, que o SRRP se comporta de acordo com o esperado e responde de maneira positiva à melhorias na infra-estrutura da rede.

Os resultados obtidos e apresentados mostraram que o SRRP é eficiente, tendo uma média de tempo de atendimento a pedidos de poucos minutos (menos de três minutos, no caso do cenário otimizado), medida que compreende desde o início do pedido, toda a negociação de qual robô vai atender ao mesmo, toda a movimentação do robô até o ponto de falha, finalizando no momento da reposição do nó.

Os resultados também demonstraram que, mesmo em simulações que estressaram o sistema, gerando falhas a uma alta taxa, de uma a cada 30 minutos, o SRRP conseguiu manter a cobertura e conectividade da rede em 90% de seu tempo (cenário otimizado), levando a crer que em um cenário real, em que as falhas normalmente acontecem em uma taxa menor, o desempenho do SRRP seria ainda melhor.

A principal dificuldade enfrentada foi o aprendizado de como se trabalhar com o NS-2 para poder implementar o SRRP e, daí, poder fazer as simulações. É conhecido no meio acadêmico que, apesar de ser a melhor escolha de simulador para redes em termos da qualidade dos resultados, o NS-2 exige muita dedicação de quem o deseja utilizar, tendo uma curva de aprendizado bem íngreme. Assim, a maior parte do tempo de desenvolvimento do trabalho

foi consumida pelo estudo do NS-2 e depois pela implementação do SRRP para o mesmo.

Alguns aspectos desse trabalho podem ser desenvolvidos e/ou aprimorados futuramente.

Como sugestões, pode-se:

- Adicionar a capacidade de perceber os obstáculos físicos da área onde a rede é implantada e considerá-los na decisão de qual robô deve atender a um pedido, já que, se para chegar a um ponto de falha um robô tiver que subir um aclave, ele terá seu consumo de energia e sua distância a ser percorrida aumentada;
- Adicionar ao líder a capacidade de tratar mais de uma falha por vez;
- Adicionar ao robô a capacidade de tratar mais de um pedido a cada movimentação. O robô poderia definir uma falha a tratar e, de acordo com um valor limite que pudesse se desviar da rota a essa falha, decidir se iria ou não tratar outra(s);
- Adicionar ao protocolo um histórico de falhas de sensores em uma célula que o líder deverá guardar. Este histórico seria utilizado também como fator para decisão de qual robô atenderá a um pedido, pois pode não ser interessante deslocar um robô de uma área que esteja tendo problemas mais frequentemente para outra que ainda não teve problema nenhum ou tenha tido poucos problemas. A negociação agora se daria além de com o robô, também com o líder da célula em que ele se encontra;
- Realizar simulações usando um posicionamento aleatório dos sensores e verificar o desempenho do protocolo. Para isso, seria necessário o uso de um algoritmo de eleição de líderes, que talvez deva ser implementado (ou até desenvolvido, adaptado), caso ainda não haja implementação para o NS-2 que se adeque ao protocolo;
- Realizar, também, simulações com uma duração maior e uma taxa de inserção de falhas menor e, dependendo dos resultados, fazer melhorias no protocolo visando aprimorar o seu desempenho.

Bibliografia

- [1] Ad hoc on-demand distance vector (aodv) routing. disponível em <<http://www.ietf.org/rfc/rfc3561.txt>>, 2003.
- [2] Sensor redeployment via robots protocol (srrp). disponível em <<https://sites.google.com/site/larissalucena/home/srrp>>, 2010.
- [3] Yazeed Al-Obaisat and Robin Braun. On wireless sensor networks: Architectures, protocols, applications, and management, 2005.
- [4] Stefano Basagni. Distributed clustering for ad hoc networks. In *ISPAN '99: Proceedings of the 1999 International Symposium on Parallel Architectures, Algorithms and Networks*, page 310, Washington, DC, USA, 1999. IEEE Computer Society.
- [5] Maxim A. Batalin and Gaurav S. Sukhatme. Spreading out: A local approach to multi-robot coverage. In *in Proc. of 6th International Symposium on Distributed Autonomous Robotic Systems*, pages 373–382, 2002.
- [6] Azzedine Boukerche, Xin Fei, and Regina B. Araujo. An energy aware coverage-preserving scheme for wireless sensor networks. In *PE-WASUN '05: Proceedings of the 2nd ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, pages 205–213, New York, NY, USA, 2005. ACM.
- [7] Guohong Cao, G. Kesidis, T. La Porta, Bin Yao, and Shashi Phoha. Purposeful mobility in tactical sensor networks, 2005.
- [8] Guohong Cao and Mukesh Singhal. A delay-optimal quorum-based mutual exclusion algorithm for distributed systems. *IEEE Trans. Parallel Distrib. Syst.*, 12(12):1256–1268, 2001.

-
- [9] Ruay-Shiung Chang and Shuo-Hung Wang. Deploying sensors for maximum coverage in sensor networks. In *IWCMC '07: Proceedings of the 2007 international conference on Wireless communications and mobile computing*, pages 220–224, New York, NY, USA, 2007. ACM.
- [10] Ioannis Chatzigiannakis, Athanasios Kinalis, and Sotiris Nikolettseas. Adaptive energy management for incremental deployment of heterogeneous wireless sensors. *Theor. Comp. Sys.*, 42(1):42–72, 2007.
- [11] Ioannis Chatzigiannakis and Sotiris Nikolettseas. A sleep-awake protocol for information propagation in smart dust networks. In *IPDPS '03: Proceedings of the 17th International Symposium on Parallel and Distributed Processing*, page 225.1, Washington, DC, USA, 2003. IEEE Computer Society.
- [12] Shafaq Chaudhry, Victor Hung, and Ratan Guha. Optimal placement of wireless sensor nodes with fault tolerance and minimal energy consumption. *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on*, pages 610–615, Oct. 2006.
- [13] D. Chen and P. K. Varshney. Qos support in wireless sensor networks: A survey. In *Proc. of the 2004 International Conference on Wireless Networks (ICWN 2004), Las Vegas, Nevada, USA*, June 2004.
- [14] S. Y. Cheung, M. H. Ammar, and M. Ahamad. The grid protocol: A high performance scheme for maintaining replicated data. *IEEE Trans. on Knowl. and Data Eng.*, 4(6):582–592, 1992.
- [15] The VINT Project A collaboratoin between researchers at UC Berkeley LBL USC/ISI and Xerox PARC. The *ns* manual, 2009.
- [16] P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, G. Sukhatme, and Csiro Ict Centre. Deployment and connectivity repair of a sensor net with a flying robot. In *In Proceedings of the 9th International Symposium on Experimental Robotics*, page 04, 2004.
- [17] Cem Ersoy Ertan Onur and Hakan Deliç. On the quality of deployment in wireless

- sensor networks. In *Proceedings of the 8th International Conference on Volume 2, Issue*, pages 547–551, 2005.
- [18] Hector Garcia-Molina and Daniel Barbara. How to assign votes in a distributed system. *J. ACM*, 32(4):841–860, 1985.
- [19] Frank Hoffmann, Michael Kaufmann, and Klaus Kriegel. The art gallery theorem for polygons with holes. In *SFCS '91: Proceedings of the 32nd annual symposium on Foundations of computer science*, pages 39–48, 1991.
- [20] Andrew Howard, Maja J Mataric, and Gaurav S. Sukhatme. An incremental self-deployment algorithm for mobile sensor networks, 2001.
- [21] Teerawat Issariyakul and Ekram Hossain. *Introduction to Network Simulator NS2*. Springer Publishing Company, Incorporated, 2008.
- [22] Holger Karl and Andreas Willig. *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & Sons, 2005.
- [23] Jize Li, Kejie Li, and Wei Zhu. Improving sensing coverage of wireless sensor networks by employing mobile robots. *Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on*, pages 899–903, Dec. 2007.
- [24] A.A.F. Loureiro, J.M.S. Nogueira, L.B. Ruiz, R.A. de Freitas Mini, E.F. Nakamura, and C.M.S. Figueiredo. Redes de sensores sem fio. In *Simposio Brasileiro de Redes de Computadores (SBRC), maio*, pages 179–226, 2003.
- [25] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 88–97, New York, NY, USA, 2002. ACM.
- [26] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M.B. Srivastava. Coverage problems in wireless ad-hoc sensor networks. *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 3:1380–1387 vol.3, 2001.

-
- [27] Ertan Onur, Cem Ersoy, and Hakan Deliç. Temporal resilience of deployment quality in surveillance wireless sensor networks. In *WCNC*, pages 3291–3296, 2008.
- [28] Curt Schurgers, Vlasios Tsiatsis, Saurabh Ganeriwal, and Mani Srivastava. Topology management for sensor networks: exploiting latency and density. In *MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 135–145, New York, NY, USA, 2002. ACM.
- [29] K. Sheu, J. Hsieh and P. Cheng. Design and implementation of mobile robot for nodes replacement in wireless sensor networks, 2008.
- [30] Sameer Tilak, Nael B. Abu-Ghazaleh, and Wendi Heinzelman. A taxonomy of wireless micro-sensor network models. *SIGMOBILE Mob. Comput. Commun. Rev.*, 6(2):28–36, 2002.
- [31] Chu-Fu Wang and Jen-Wen Ding. The optimum sensor redeployment scheme using the most frangible clusters set. *Comput. Commun.*, 31(14):3492–3502, 2008.
- [32] Guiling Wang, Guohong Cao, Piotr Berman, and Thomas F. Laporta. A bidding protocol for deploying mobile sensors. In *in Proceedings of IEEE ICNP*, pages 315–324, 2003.
- [33] Guiling Wang, Guohong Cao, and Thomas F. La Porta. Movement-assisted sensor deployment. *IEEE Transactions on Mobile Computing*, 5(6):640–652, 2006.
- [34] Xiaoling Wu, Jinsung Cho, Brian J. d’Auriol, Sungyoung Lee, and Hee Yong Youn. Self-deployment of mobile nodes in hybrid sensor networks by ahp. In *UIC*, pages 663–672, 2007.
- [35] Guoliang Xing, Xiaorui Wang, Yuanfang Zhang, Chenyang Lu, Robert Pless, and Christopher Gill. Integrated coverage and connectivity configuration for energy conservation in sensor networks. *ACM Trans. Sen. Netw.*, 1(1):36–72, 2005.
- [36] Kenan Xu. *Device Deployment Strategies for Large-scale Wireless Sensor Networks*. PhD thesis, Queen’s University, 2008.

-
- [37] H. Yang and B. Sikdar. A protocol for tracking mobile targets using sensor networks. Anchorage, AK, 05/2003 2003.
- [38] Yinying Yang and Mihaela Cardei. Movement-assisted sensor redeployment scheme for network lifetime increase. In *MSWiM '07: Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*, pages 13–20, New York, NY, USA, 2007. ACM.
- [39] Chee yee Chong, Ieee, Srikanta P. Kumar, and Senior Member. Sensor networks: evolution, opportunities, and challenges. In *Proceedings of the IEEE*, pages 1247–1256, 2003.
- [40] Weili Wu Yingshu Li, My T. Thai. *Wireless Sensor Networks and Applications*. Springer, 2008.