

**Pontifícia Universidade Católica do Rio Grande do Sul**  
**Faculdade de Informática**  
**Programa de Pós-Graduação em Ciência da Computação**

**Mecanismos de Coordenação**  
**Ator-Ator em Redes de Sensores**  
**e Atores sem Fio**

Marcelo da Silva Strzykalski

**Dissertação apresentada como**  
**requisito parcial à obtenção do**  
**grau de mestre em Ciência da**  
**Computação**

Orientador: Prof. Dr. Luiz Gustavo Leão Fernandes

Porto Alegre  
2009

## Dados Internacionais de Catalogação na Publicação (CIP)

S928m Strzykalski, Marcelo da Silva.  
Mecanismos de coordenação ator-ator em redes de sensores e atores sem fio / Marcelo da Silva Strzykalski. – Porto Alegre, 2009.  
106 f.

Diss. (Mestrado) – Fac. de Informática, PUCRS.  
Orientador: Prof. Dr. Luiz Gustavo Leão Fernandes

1. Informática. 2. Redes de Sensores e Atores sem Fio. 3. Coordenação Ator-Ator (Informática). 4. Alocação de Tarefas. 5. Avaliação de Desempenho (Informática). I. Fernandes, Luiz Gustavo Leão. II. Título.

CDD 004.68

**Ficha Catalográfica elaborada pelo  
Setor de Tratamento da Informação da BC-PUCRS**



Pontifícia Universidade Católica do Rio Grande do Sul  
FACULDADE DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

## TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada "**Mecanismos de Coordenação Ator-Ator em Redes de Sensores e Atores sem Fio**", apresentada por Marcelo da Silva Strzykalski, como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, Processamento Paralelo e Distribuído, aprovada em 26/01/09 pela Comissão Examinadora:

Prof. Dr. Luiz Gustavo Leão Fernandes -  
Orientador

PPGCC/PUCRS

Prof. Dr. Avelino Francisco Zorzo -

PPGCC/PUCRS

Prof. Dr. Luciano Paschoal Gaspar -

UFRGS

Homologada em 05/05/09, conforme Ata No. 007 pela Comissão Coordenadora.

Prof. Dr. Fernando Gehm Moraes  
Coordenador.

PUCRS

### Campus Central

Av. Ipiranga, 6681 - P32 - sala 507 - CEP: 90619-900

Fone: (51) 3320-3611 - Fax (51) 3320-3621

E-mail: [ppgcc@pucrs.br](mailto:ppgcc@pucrs.br)

[www.pucrs.br/facin/pos](http://www.pucrs.br/facin/pos)

## **Agradecimentos**

À minha família, que sempre me apoiou, mesmo que em não raras oportunidades eu lhes tenha ignorado completamente. Da mesma forma, agradeço aos meus amigos pela compreensão, pois estive ausente durante a quase totalidade do período de realização do presente trabalho.

Um agradecimento especial ao meu orientador, Fernando Luís Dotti, pela paciência dispendida. Gostaria de agradecer ao professor Avelino Francisco Zorzo pelas importantes observações realizadas durante as etapas intermediárias ao desenvolvimento dessa dissertação, bem como ao meu co-orientador, o professor Luiz Gustavo Leão Fernandes, pela atenção e pela presteza que viabilizaram a execução das tarefas associadas à etapa de finalização do presente trabalho.

## Resumo

Redes de sensores e atores sem fio são sistemas embarcados distribuídos, os quais são constituídos por um grupo de nós sensores e atores que se comunicam de forma espontânea por meio de enlaces sem fio. Os sensores são dispositivos de pequeno tamanho e de baixo custo que possuem limitada autonomia energética, capacidade computacional e de comunicação, os quais são responsáveis pela coleta de informações sobre o mundo físico. Os atores são dispositivos que possuem maior autonomia energética, capacidade computacional e de comunicação, os quais executam ações no ambiente controlado. Sensores e atores colaboram entre si para monitorar e para reagir a variações nas condições ambientais dentro de limites temporais especificados. Os sensores realizam o sensoriamento dos fenômenos físicos e reportam os dados sensorizados para os atores, enquanto que os atores se coordenam entre si para tomar decisões a respeito da forma mais apropriada de atuar no mundo físico em resposta aos eventos detectados pelos sensores. A coordenação ator-ator consiste na seleção do ator ou conjunto de atores que devem executar a ação no ambiente, os quais necessitam se coordenar entre si da melhor forma possível para maximizar o desempenho global na execução de tarefas. De acordo com essa definição, os nós atores devem se coordenar de forma autônoma para que decisões sejam tomadas em tempo hábil, e que como consequência, ações possam ser executados no ambiente controlado pelo sistema. No entanto, poucos trabalhos na literatura propõem mecanismos direcionados para tal cenário. Dessa forma, a principal contribuição desse trabalho reside na identificação de estratégias aplicáveis ao desenvolvimento de mecanismos para a coordenação ator-ator, bem como na implementação e na avaliação de desempenho de mecanismos derivados de tais estratégias.

**Palavras-chave:** Redes de Sensores e Atores sem Fio, Coordenação, Alocação de Tarefas, Redes de Contratos, Inteligência Coletiva

## **Abstract**

Wireless sensor and actor networks are distributed, embedded systems comprised of sensors and actor nodes communicating in an ad hoc manner through wireless links. Sensors are small, low-cost devices with limited computing and communication capabilities responsible for collecting information about the physical world. Actors are devices with longer battery life and better computing and communication capabilities that perform actions in the controlled environment. Sensors and actors collaborate to monitor and react to changes in the surrounding world within specified times. Sensors perform sensing and report sensed data to the actors, while the actors coordinate with each other to take decisions on the most appropriate way to act in the physical world in response to the events detected by sensors. Actor-actor coordination consists on the selection of the actor or set of actors that should perform the action in the environment, coordinating with each other in the best way to maximize the overall task execution performance. In accordance to this definition, actor nodes should coordinate autonomously to take decisions within timing constraints, and as a consequence, actions would be performed in the controlled system. However, few works in the literature propose mechanisms focused on such situations. Thus, the main contribution of this work resides in the identification of strategies applied to the development of actor-actor coordination mechanisms, as well as in the implementation and performance evaluation of mechanisms derived from such strategies.

**Keywords:** Wireless Sensor and Actors Networks, Coordination, Task Allocation, Contract Nets, Swarm Intelligence

## Lista de Figuras

Figura 1	Arquitetura de uma Rede Móvel <i>Ad Hoc</i> . . . . .	19
Figura 2	Arquitetura de uma Rede Local sem Fio com Infraestrutura . . . . .	20
Figura 3	Arquitetura de um Nó Sensor . . . . .	21
Figura 4	Arquitetura de uma Rede de Sensores sem Fio . . . . .	21
Figura 5	Arquitetura de um Nó Ator . . . . .	23
Figura 6	Arquitetura de uma Rede de Sensores e Atores sem Fio . . . . .	23
Figura 7	Arquitetura de uma Rede de Sensores e Atores sem Fio Automatizada . . . . .	24
Figura 8	Arquitetura de uma Rede de Sensores e Atores sem Fio Semi-Automatizada . . . . .	24
Figura 9	Arquitetura de um Nó Ator em uma Rede do Tipo SANET . . . . .	25
Figura 10	Arquitetura de um Sistema de Tempo Real . . . . .	26
Figura 11	Diagrama de Sequência em UML da Versão Base do Protocolo de Redes de Contratos . . . . .	34
Figura 12	Diagrama de Sequência em UML da Versão Estendida do Protocolo de Redes de Contratos . . . . .	35
Figura 13	Modelo de Mecanismo Regulador do Comportamento Laboral Baseado em Resultados Obtidos com Abelhas . . . . .	42
Figura 14	Diagrama de Blocos de um Sistema de Controle a Malha Aberta . . . . .	54
Figura 15	Diagrama de Blocos de um Sistema de Controle a Malha Fechada . . . . .	55
Figura 16	Arquitetura de um Sistema Mecatrônico . . . . .	55
Figura 17	Arquitetura de uma Unidade de Atuação . . . . .	56
Figura 18	Arquitetura do <i>Framework</i> de Simulação . . . . .	56
Figura 19	Diagrama de Classes em UML do WSANSim . . . . .	58
Figura 20	Desempenho dos Mecanismos de Coordenação Ator-Ator para o Modelo de Mobilidade Estático em Redes com Diferentes Números de Atores (a) Tempo de Resposta Médio (b) Taxa de Sucesso no Atendimento de Limitações Temporais (c) Taxa de Sucesso na Cobertura das Áreas de Eventos (d) Energia Residual Média . . . . .	75
Figura 21	Desempenho dos Mecanismos de Coordenação Ator-Ator para o Modelo de Mobilidade <i>Random Walk</i> em Redes com Diferentes Números de Atores (a) Tempo de Resposta Médio (b) Taxa de Sucesso no Atendimento de Limitações Temporais (c) Taxa de Sucesso na Cobertura das Áreas de Eventos (d) Energia Residual Média . . . . .	78
Figura 22	Desempenho dos Mecanismos de Coordenação Ator-Ator para o Modelo de Propagação de Rádio <i>Free Space</i> em Redes com Diferentes Números de Atores (a) Tempo de Resposta Médio (b) Taxa de Sucesso no Atendimento de Limitações Temporais (c) Taxa de Sucesso na Cobertura das Áreas de Eventos (d) Energia Residual Média . . . . .	82

Figura 23	Desempenho dos Mecanismos de Coordenação Ator-Ator para o Modelo de Propagação de Rádio <i>Shadowing</i> em Redes com Diferentes Números de Atores (a) Tempo de Resposta Médio (b) Taxa de Sucesso no Atendimento de Limitações Temporais (c) Taxa de Sucesso na Cobertura das Áreas de Eventos (d) Energia Residual Média . . . . .	85
-----------	--	----



## Lista de Tabelas

Tabela 1	Notações Empregadas nos Mecanismos de Coordenação Ator-Ator Sugeridos . . . . .	29
Tabela 2	Parâmetros de Simulação do Estudo de Caso com o WSANSim . . . . .	66
Tabela 3	Resultados Experimentais do Estudo de Caso com o WSANSim . . . . .	67
Tabela 4	Parâmetros Genéricos de Simulação dos Mecanismos de Coordenação Ator-Ator . . . . .	70
Tabela 5	Parâmetros Específicos de Simulação - Coordenação Ator-Ator Baseada no Protocolo de Redes de Contratos . . . . .	71
Tabela 6	Parâmetros Específicos de Simulação - Coordenação Ator-Ator Baseada em Extensões ao Protocolo de Redes de Contratos . . . . .	71
Tabela 7	Parâmetros Específicos de Simulação - Coordenação Ator-Ator Baseada no Modelo de Limiar de Resposta . . . . .	71
Tabela 8	Parâmetros Específicos de Simulação - Coordenação Ator-Ator Baseada em Extensões ao Modelo de Limiar de Resposta . . . . .	71
Tabela 9	Parâmetros Específicos de Simulação - Modelo de Mobilidade Empregado pelos Nós Atores . . . . .	71
Tabela 10	Parâmetros Específicos de Simulação - Modelo de Propagação de Rádio <i>Shadowing</i> . . . . .	71
Tabela 11	Siglas Empregadas na Identificação dos Mecanismos Propostos para a Coordenação Ator-Ator . . . . .	72
Tabela 12	Medidas de Variabilidade da Quantidade de Energia Residual do Conjunto de Nós Atores para os Mecanismos de Coordenação Ator-Ator Baseados no Modelo de Mobilidade Estático em Redes com Diferentes Números de Atores . . . . .	77
Tabela 13	Medidas de Variabilidade da Quantidade de Energia Residual do Conjunto de Nós Atores para os Mecanismos de Coordenação Ator-Ator Baseados no Modelo de Mobilidade <i>Random Walk</i> em Redes com Diferentes Números de Atores . . . . .	80
Tabela 14	Medidas de Variabilidade da Quantidade de Energia Residual do Conjunto de Nós Atores para os Mecanismos de Coordenação Ator-Ator Baseados no Modelo de Propagação de Rádio <i>Free Space</i> em Redes com Diferentes Números de Atores . . . . .	84
Tabela 15	Medidas de Variabilidade da Quantidade de Energia Residual do Conjunto de Nós Atores para os Mecanismos de Coordenação Ator-Ator Baseados no Modelo de Propagação de Rádio <i>Shadowing</i> em Redes com Diferentes Números de Atores . . . . .	87

## Lista de Siglas

<b>MANET</b>	<i>Mobile Ad Hoc Networks</i>	13
<b>WSN</b>	<i>Wireless Sensor Networks</i>	13
<b>WSAN</b>	<i>Wireless Sensor and Actor Networks</i>	13
<b>MEMS</b>	<i>Micro Electro-Mechanical Systems</i>	19
<b>SANET</b>	<i>Sensor Actor Networks</i>	24
<b>SAT</b>	<i>Single-Actor Task</i>	28
<b>MAT</b>	<i>Multi-Actor Task</i>	28
<b>GPS</b>	<i>Global Positioning System</i>	29
<b>CNP</b>	<i>Contract-Net Protocol</i>	33
<b>pdns</b>	<i>Parallel/Distributed Network Simulator</i>	50
<b>ACA</b>	<i>Autonomous Component Architecture</i>	51
<b>SENSE</b>	<i>Sensor Network Simulator and Emulator</i>	53
<b>COST</b>	<i>Component Oriented Simulation</i>	53
<b>SENS</b>	<i>Sensor, Environment and Network Simulator</i>	53
<b>ODE</b>	<i>Open Dynamics Engine</i>	53
<b>LEACH</b>	<i>Low Energy Adaptive Clustering Hierarchy Protocol</i>	59
<b>ASHRAE</b>	<i>American Society of Heating, Refrigerating and Air-Conditioning Engineers</i>	65
<b>HVAC</b>	<i>Heating, Ventilation and Air Conditioning</i>	65
<b>BPRC</b>	Mecanismo de Coordenação Ator-Ator Baseado no Protocolo de Redes de Contratos	72
<b>BEPRC</b>	Mecanismo de Coordenação Ator-Ator Baseado em Extensões ao Protocolo de Redes de Contratos	72
<b>BMLR</b>	Mecanismo de Coordenação Ator-Ator Baseado no Modelo de Limiar de Resposta	72
<b>BEMLR</b>	Mecanismo de Coordenação Ator-Ator Baseado em Extensões ao Modelo de Limiar de Resposta	72
<b>RCR</b>	<i>Real-time Coordination and Routing</i>	93

# Sumário

<b>1</b>	<b>Introdução</b>	<b>13</b>
1.1	Contribuições	16
1.2	Organização do Texto	16
<b>2</b>	<b>Redes de Sensores e Atores</b>	<b>18</b>
2.1	Redes Móveis <i>Ad Hoc</i>	18
2.2	Redes de Sensores sem Fio	19
2.3	Redes de Sensores e Atores sem Fio	22
<b>3</b>	<b>Mecanismos de Coordenação Ator-Ator</b>	<b>27</b>
3.1	Notações Empregadas na Apresentação dos Mecanismos	28
3.2	Pressuposições	29
3.3	Modelo de Sistema	30
3.4	Baseados em Abordagens Econômicas	31
3.4.1	O Protocolo de Redes de Contratos	34
3.4.2	Baseado no Protocolo de Redes de Contratos	36
3.4.3	Baseado em Extensões ao Protocolo de Redes de Contratos	37
3.4.4	Trabalhos Relacionados	38
3.5	Baseados em Abordagens Biológicas	39
3.5.1	O Modelo de Limiar de Resposta	41
3.5.2	Baseado no Modelo de Limiar de Resposta	45
3.5.3	Baseado em Extensões ao Modelo de Limiar de Resposta	46
3.5.4	Trabalhos Relacionados	48
<b>4</b>	<b>WSANSim: Um Ambiente de Simulação para WSN</b>	<b>49</b>
4.1	Definição do Problema	49
4.2	Trabalhos Relacionados	49
4.2.1	ns-2	50
4.2.2	pdns	50
4.2.3	SensorSim	51
4.2.4	J-Sim	51
4.2.5	Ptolemy	52
4.2.6	Mannasim	52
4.2.7	SENSE	53
4.2.8	SENS	53
4.2.9	Sidh	53
4.2.10	BAKARA	53
4.3	Arquitetura	54
4.4	Implementação	57

4.5	Estudo de Caso . . . . .	64
4.5.1	Modelo de Simulação . . . . .	65
4.5.2	Resultados . . . . .	65
<b>5</b>	<b>Avaliação de Desempenho . . . . .</b>	<b>68</b>
5.1	Modelo de Simulação . . . . .	69
5.2	Requisitos Avaliados . . . . .	72
5.3	Medidas de Desempenho . . . . .	72
5.4	Resultados Experimentais . . . . .	73
5.4.1	Efeito do Modelo de Mobilidade . . . . .	74
5.4.2	Efeito do Modelo de Propagação de Rádio . . . . .	81
<b>6</b>	<b>Conclusão . . . . .</b>	<b>89</b>
6.1	Contribuições . . . . .	90
6.2	Trabalhos Futuros . . . . .	90
6.2.1	Desenvolvimento de Extensões para o Protocolo de Redes de Contratos . . . . .	90
6.2.2	Coordenação Ator-Ator Tolerante a Falhas . . . . .	91
6.2.3	Mecanismos para a Coordenação Sensor-Sensor e Sensor-Ator . . . . .	93
6.2.4	Modelos de Mobilidade para Atores Móveis . . . . .	94
6.2.5	Migração do WSANSim para Outros Ambientes de Simulação . . . . .	94
6.2.6	Implementação de Técnicas de Emulação no ns-2 . . . . .	95
6.2.7	Implementação de <i>Frontend</i> Gráfico para o WSANSim . . . . .	95
	<b>Referências . . . . .</b>	<b>96</b>

# 1 Introdução

Uma rede móvel *ad hoc* (ou *Mobile Ad Hoc Networks* - MANET) é uma rede sem fio auto-configurável, a qual é constituída por nós que atuam como roteadores e que podem mover-se de forma arbitrária [66]. Em uma MANET não existe uma infraestrutura de comunicação fixa, diferentemente do que ocorre em uma rede sem fio convencional [144]. Uma rede de sensores sem fio (ou *Wireless Sensor Networks* - WSN) é uma especialização de uma rede móvel *ad hoc*, a qual é constituída por nós sensores que são distribuídos espacialmente em uma dada região geográfica e cuja finalidade consiste na detecção de eventos do ambiente por meio do uso de dispositivos de sensoriamento tais como de temperatura, de pressão, etc [8]. Particularmente, em tal classe de rede os eventos capturados pelos nós sensores são enviados para o nó escoa-douro, o qual centraliza a coleta e o processamento dos dados sensorizados do ambiente. Por sua vez, uma rede de sensores e atores sem fio (ou *Wireless Sensor and Actor Networks* - WSAN) é uma extensão de uma rede de sensores, a qual é composta adicionalmente por nós atores e seus respectivos dispositivos de atuação. Como resultado, tal acréscimo possibilita tanto a monito-ria do ambiente quanto o desempenho de ações de controle sobre esse. A literatura menciona diversos cenários de uso de tal tecnologia [7], dentre os quais cabe ressaltar monitoria ambi-ental, automação doméstica, automação industrial, automação predial e automação de tarefas relacionadas com necessidades militares.

Uma rede de sensores e atores sem fio possui requisitos de tempo real [7]. Dependendo da aplicação, é necessário que ocorra uma resposta rápida a um evento capturado por um nó sensor. Isso implica no fato de que uma vez que um evento tenha sido detectado, os nós atores devem atuar no ambiente dentro de um intervalo de tempo pré-definido. Por conseguinte, uma WSAN pode ser concebida como um sistema de controle distribuído que deve reagir em tempo real a estímulos coletados do ambiente [108]. Por essa razão, a comunicação e a coordenação em tempo real é uma preocupação importante no contexto de uma WSAN, o que pode garantir a execução de ações corretas no ambiente dentro de intervalos de tempo fornecidos pelos requi-sitos da aplicação [7]. A eficiência energética da rede de comunicação também é crucial, já que os nós sensores possuem recursos de processamento e memória limitados, bem como dispositi-vos de fornecimento de energia que possuem um tempo de vida finito. Em acréscimo a tais restrições, os algoritmos projetados para essa categoria de rede sem fio devem ser escaláveis e localizados, uma vez que o número de nós sensores espalhados no campo de sensores pode ser arbitrariamente alto [183].

Diferentemente do que ocorre em uma WSN, na qual existe uma entidade central que desem-penha as funções de coordenação e de coleta dos dados, em uma WSAN surge o fenômeno da

coordenação. De acordo com a literatura, constata-se a existência dos seguintes tipos de coordenação: coordenação sensor-sensor, coordenação sensor-ator e coordenação ator-ator [7] [183]. Esse modelo de coordenação em três níveis é descrito a seguir:

- Coordenação sensor-sensor: o principal objetivo dessa modalidade de coordenação consiste em viabilizar a coleta de informações a respeito dos fenômenos físicos de uma forma efetiva e energeticamente eficiente, de modo que o tempo de vida da rede possa ser maximizado. Assim, a coordenação sensor-sensor implica na construção de um mecanismo que possibilite a clusterização dos nós sensores e que periodicamente ligue e desligue o subsistema de comunicação de tais nós.
- Coordenação sensor-ator: essa modalidade deve possibilitar a transmissão dos eventos capturados pelos nós sensores aos nós atores, de maneira tal que a latência e o consumo energético sejam minimizados. Portanto, a coordenação sensor-ator consiste na definição de um mecanismo que permita aos sensores selecionarem os atores que receberão os eventos capturados do ambiente.
- Coordenação ator-ator: em tal modalidade, os nós atores devem se coordenar entre si com o objetivo de minimizar o tempo de resposta e o dispêndio de energia na execução das ações de controle. Logo, a coordenação ator-ator traduz-se na especificação de um mecanismo que viabilize a determinação do ator ou do conjunto de atores mais aptos para o desempenho de uma dada tarefa no ambiente controlado.

Dessa forma, o problema da coordenação ator-ator consiste na atribuição ótima de tarefas a um ator individual ou a um grupo de atores [7], o qual é similar em sua essência ao problema da alocação de tarefas no âmbito dos sistemas distribuídos em geral [108]. Em um sistema distribuído, um processo é composto por um conjunto de tarefas que interagem entre si por meio de uma rede de comunicação [152]. No âmbito desse tipo de sistema computacional, o problema da alocação de tarefas consiste na determinação da política ótima de atribuição das tarefas que constituem um processo individual ao conjunto de nós processadores do sistema, com o objetivo de minimizar o *overhead* que é introduzido a partir da comunicação entre os processos [46] [64] [65]. Portanto, o problema da alocação de tarefas clássico em um sistema distribuído de propósito geral consiste no mapeamento das tarefas que devem ser executadas aos nós processadores do sistema por meio do emprego de algum critério de otimização [160]. Contudo, uma WSN é um sistema distribuído cuja finalidade principal reside na interação com o ambiente. Logo, o problema da alocação de tarefas nesse contexto consiste mais especificamente na identificação da melhor forma possível de se interagir com o ambiente que está sendo monitorado por meio de ações de controle desempenhadas pelos atores, de tal modo que limitações temporais possam ser atendidas e que o dispêndio energético na execução de tais ações possa ser minimizado.

No contexto da coordenação ator-ator, os atores devem colaborar de forma autônoma para que decisões sejam tomadas em tempo hábil, e que como consequência, ações possam ser executadas no ambiente controlado. Logo, pode-se conduzir uma investigação a respeito de que tipos de estratégias são capazes de dotar tais entidades de capacidade decisória para executar as ações necessárias nesse ambiente, e ainda prover o atendimento das restrições temporais impostas. Todavia, poucos trabalhos na literatura propõem soluções para o problema da coordenação ator-ator, sendo que estes não apresentam resultados experimentais que possam comprovar a efetividade dos mecanismos propostos, ou dependem da existência de um ambiente centralizado para a resolução de problemas modelados a partir do emprego de técnicas de programação matemática ou não levam em conta o grau de aptidão corrente de cada ator para a execução da ação de controle sobre o ambiente.

De acordo com publicações recentes no âmbito dos sistemas multi-agentes e dos sistemas com múltiplos robôs, verifica-se uma certa proeminência de abordagens para a alocação dinâmica de tarefas baseadas em metáforas econômicas, as quais se inspiram no funcionamento das economias de mercado [29] [59] [72] [74] [106], ou em metáforas biológicas, as quais se inspiram no comportamento coletivo das sociedades de insetos [3] [47] [62]. Usualmente, as abordagens econômicas se baseiam em um algum mecanismo de negociação, enquanto que as abordagens biológicas se baseiam no paradigma *Swarm Intelligence* [21], o qual reusa os princípios que norteiam o comportamento adaptativo e inteligente dos sistemas naturais para a implementação de sistemas artificiais. Cabe enfatizar que o emprego de tais metáforas no desenvolvimento de algoritmos para a alocação dinâmica de tarefas encontra justificativa no fato de que tais abordagens permitem a um sistema computacional operar de forma coordenada por meio do uso de conhecimento imperfeito a respeito das condições ambientais, o que lhe permite adaptar-se de forma automática a mudanças no ambiente ou a perturbações em seus componentes individuais [81] [111] [24].

Visto que mecanismos de coordenação ator-ator devem viabilizar a alocação dinâmica de tarefas entre atores que estão inseridos em ambientes que possuem certo grau de mutabilidade, é razoável supor que se possa reusar tais abordagens para a elaboração desses mecanismos. Portanto, o foco principal da presente dissertação consiste no reuso de tais estratégias para a especificação e para a implementação de algoritmos distribuídos que possibilitem tal comportamento colaborativo. Frequentemente, emprega-se a técnica de simulação para a avaliação de desempenho de tais algoritmos. No entanto, constata-se que não existe um ambiente de simulação para WSAAN bem estabelecido na comunidade de pesquisa. Por tal motivo, concebeu-se o WSANSim, o qual foi implementado como uma extensão do simulador de redes ns-2.

## 1.1 Contribuições

A investigação e o desenvolvimento de mecanismos para a coordenação ator-ator em redes de sensores e atores sem fio trazem as seguintes contribuições:

- A identificação de estratégias que possam ser aplicáveis na elaboração de tais mecanismos. Foram identificadas duas estratégias, as quais são baseadas no comportamento dos mecanismos de mercado e no comportamento colaborativo dos insetos.
- A especificação e a implementação de mecanismos para a coordenação ator-ator baseados nos paradigmas econômico e biológico.
- A especificação de uma arquitetura genérica para a simulação de redes de sensores e atores sem fio.
- A implementação dessa arquitetura por meio de extensões ao simulador de redes ns-2, bem como a apresentação de uma aplicação do ambiente de simulação implementado em um estudo de caso prático o qual compara os resultados obtidos nesse com resultados existentes na literatura.
- A avaliação de desempenho dos mecanismos implementados para a coordenação ator-ator, tendo em vista medidas de desempenho tais como o tempo de resposta do sistema, a taxa de sucesso no atendimento de restrições temporais, a taxa de sucesso na cobertura das áreas de eventos e a quantidade de energia residual do sistema, em redes nas quais o número de nós é variável.

## 1.2 Organização do Texto

Este texto está organizado em seis capítulos. O Capítulo 2 caracteriza de forma sucinta as redes móveis *ad hoc*, as redes de sensores e as redes de sensores e atores sem fio.

O Capítulo 3 apresenta as estratégias que foram identificadas para o desenvolvimento de mecanismos focados na coordenação ator-ator, as quais se dividem em duas categorias: as que se baseiam no emprego de princípios econômicos e as que se baseiam no uso de princípios biológicos. Além disso, discutem-se os mecanismos de coordenação ator-ator que foram implementados a partir das estratégias que foram identificadas. Adicionalmente, compara os mecanismos elaborados com trabalhos mencionados na literatura.

No Capítulo 4 é apresentado o WSANSim, um *framework* que foi proposto com o objetivo de possibilitar a simulação de redes de sensores e atores sem fio. Nesse capítulo, descreve-se a arquitetura genérica de simulação que foi sugerida para tal classe de rede, discute-se a implementação dessa arquitetura no simulador de redes ns-2 e apresenta-se um estudo de caso



prático no qual se emprega o ambiente de simulação proposto para a comparação dos resultados obtidos com resultados publicados na literatura.

O Capítulo 5 discute os resultados que foram obtidos a partir da simulação dos mecanismos de coordenação ator-ator propostos, os quais foram implementados no WSANSim.

O Capítulo 6 apresenta as conclusões obtidas a partir da simulação dos mecanismos de coordenação ator-ator especificados e implementados. Discutem-se ainda as contribuições que foram trazidas pelo presente trabalho bem como mencionam-se trabalhos futuros a serem conduzidos.

## 2 Redes de Sensores e Atores

Redes de sensores sem fio são sistemas computacionais auto-organizáveis formados por nós que cooperam para a criação de uma infra-estrutura temporária de rede [8]. Tal classe de rede sem fio pode introduzir diversas vantagens em cenários que exijam a monitoria das condições ambientais. Uma WSN possibilita a conectividade dos nós da rede sem que se faça necessário a existência de um elemento de controle central. Essa pode operar de forma autônoma em ambientes que são insalubres e perigosos, e pode ser facilmente reconfigurável para o atendimento de requisitos específicos de uma dada aplicação.

Uma rede de sensores é um tipo especial de rede móvel *ad hoc* [8]. No entanto, uma MANET possui como função básica prover suporte a comunicação entre elementos computacionais que trocam dados diretamente entre si, e que individualmente podem estar executando tarefas distintas. Por outro lado, uma WSN tende a executar uma função colaborativa na qual os nós sensores coletam dados que são processados por um nó especial denominado *sink node*.

Recentemente, a partir de uma tentativa de se incorporar as redes de sensores às diversas atividades necessárias para o funcionamento da maquinaria social humana, reconheceu-se que seria benéfico o acréscimo de atores e atuadores às redes de sensores [76]. Os atuadores são dispositivos simples, os quais são programados para executar ações imediatas em resposta aos dados coletados pelas redes de sensores, enquanto que os atores são entidades de rede sofisticadas, os quais em adição a execução de ações, podem interagir de forma duradoura com o ambiente. Nesse contexto, cabe destacar que um atuador é um dos elementos que constituem a arquitetura de um nó ator. A tipologia de rede resultante desses acréscimos às redes de sensores sem fio é comumente referenciada na literatura como redes de sensores e atores sem fio.

Tais tipologias de redes *ad hoc* serão descritas de forma mais detalhada nas seções a seguir.

### 2.1 Redes Móveis *Ad Hoc*

Uma rede móvel *ad hoc* é uma rede sem fio que pode ser formada espontaneamente e que é independente de qualquer infraestrutura de rede convencional ou administração centralizada [2] (ver Figura 1), ao contrário do que acontece em uma rede sem fio baseada na arquitetura centralizada do sistema de telefonia celular (ou rede local sem fio com infraestrutura, a qual pode ser visualizada na Figura 2), na qual existe um *access point* que conecta os dispositivos ao *backbone* da rede fixa. Mais especificamente, uma MANET é um tipo de sistema autônomo

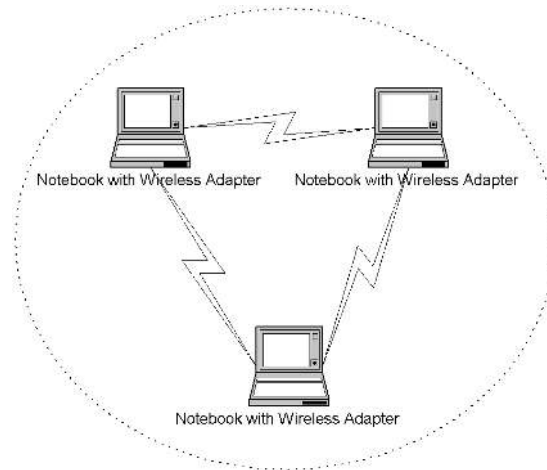


Figura 1 – Arquitetura de uma Rede Móvel *Ad Hoc*

composto por uma coleção de dispositivos móveis que possuem interfaces de rede sem fio, os quais se conectam por meio de uma topologia dinâmica e arbitrária [20]. Cada nó possui um alcance de transmissão limitado e pode se comunicar diretamente com os nós que estão dentro do seu alcance de transmissão (do qual deriva a relação de vizinhança entre os nós).

A comunicação com os outros nós da rede é viabilizada pelo roteamento de pacotes através dos nós vizinhos. Essa classe de rede sem fio se caracteriza pela limitada largura de banda, pela limitada quantidade de energia disponível em cada nó e pela limitada distância de comunicação entre os nós [66]. Além disso, cada nó está sujeito a freqüentes modificações da topologia da rede e ao conseqüente particionamento dessa, bem como ao funcionamento em modo de economia de energia [139].

## 2.2 Redes de Sensores sem Fio

Recentes avanços tecnológicos que ocorreram nos últimos anos na área de microprocessadores, novos materiais de sensoriamento, micro sistemas eletro-mecânicos (MEMS - Micro Electro-Mechanical Systems ) e comunicação sem fio têm estimulado o desenvolvimento e o uso de sensores "inteligentes" em áreas ligadas a processos físicos, químicos, biológicos, dentre outros [8]. É comum que se tenha num único *chip* vários sensores, os quais são controlados pela lógica do circuito integrado, com uma interface de comunicação sem fio. A arquitetura de um nó sensor é usualmente composta pelos seguintes componentes (ver Figura 3 - adaptada de [7]) [131]:

- Subsistema de computação: consiste de um microprocessador (ou microcontrolador), o qual é responsável pelo controle dos sensores e pela execução dos protocolos de comunicação. Esse usualmente pode funcionar em diversos modos operacionais, para propósitos

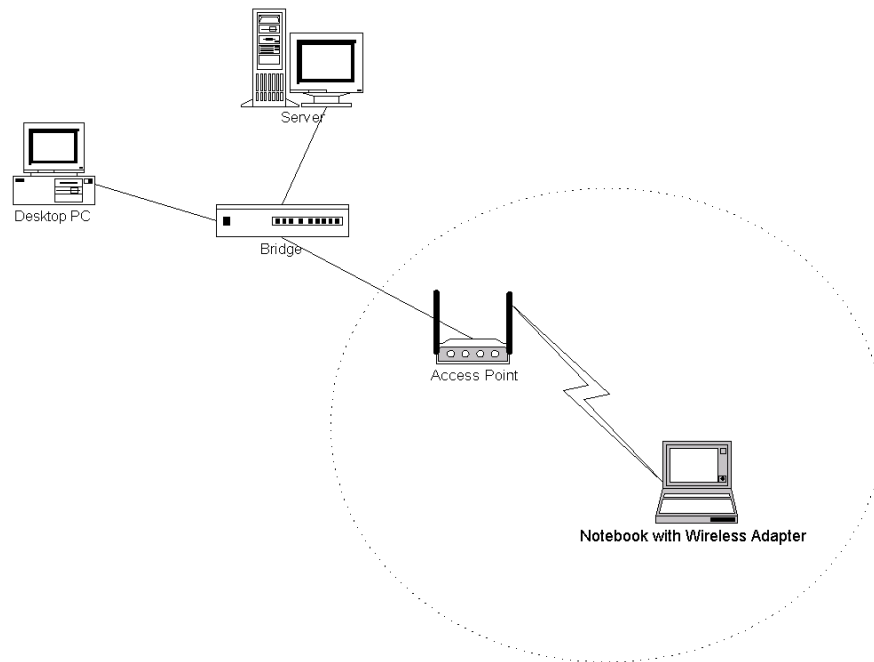


Figura 2 – Arquitetura de uma Rede Local sem Fio com Infraestrutura

de gerenciamento de energia.

- Subsistema de comunicação: consiste em um dispositivo de rádio de curto alcance, o qual é empregado para a comunicação com os nós vizinhos e com o mundo exterior. Esse pode transitar entre os modos de operação *transmit*, *receive*, *idle* e *sleep*. É importante que se efetue o *shutdown* completo desse dispositivo ao invés de colocá-lo em modo *idle* quando esse não está nos estados *transmit* e *receive*, devido ao alto consumo de energia associado a esses modos de operação [162] [44].
- Subsistema de sensoriamento: consiste de um grupo de sensores que conectam o nó sensor ao mundo externo. Cada nó pode ser equipado com uma variedade de sensores, tais como: acústico, sísmico, infravermelho, temperatura, pressão, etc.
- Subsistema de alimentação: consiste de uma bateria, a qual fornece energia para todos os subsistemas de *hardware* que constituem o nó sensor.

Uma rede de sensores sem fio é um tipo especial de rede móvel *ad hoc* distribuída, a qual se configura de forma espontânea para o monitoramento de fenômenos do mundo real [8]. Tais sensores podem ser conectados estática ou dinamicamente ao ambiente que deve ser monitorado; além disso, podem ser instalados em ambientes inaccessíveis, hostis ou perigosos. Usualmente, os nós sensores são espalhados no campo de sensores de acordo com a Figura 4 (adaptada de [8]), sendo que esses devem se coordenar entre si para a geração de informações a respeito do ambiente. Essas podem então ser transportadas em direção ao ponto de acesso, o qual é o elemento por meio do qual a rede comunica-se com outras redes ou com um ou mais

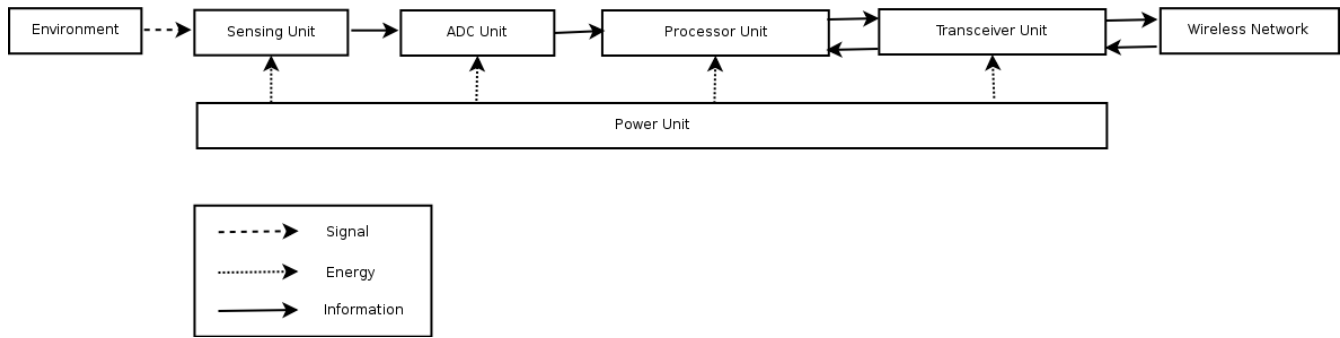


Figura 3 – Arquitetura de um Nó Sensor

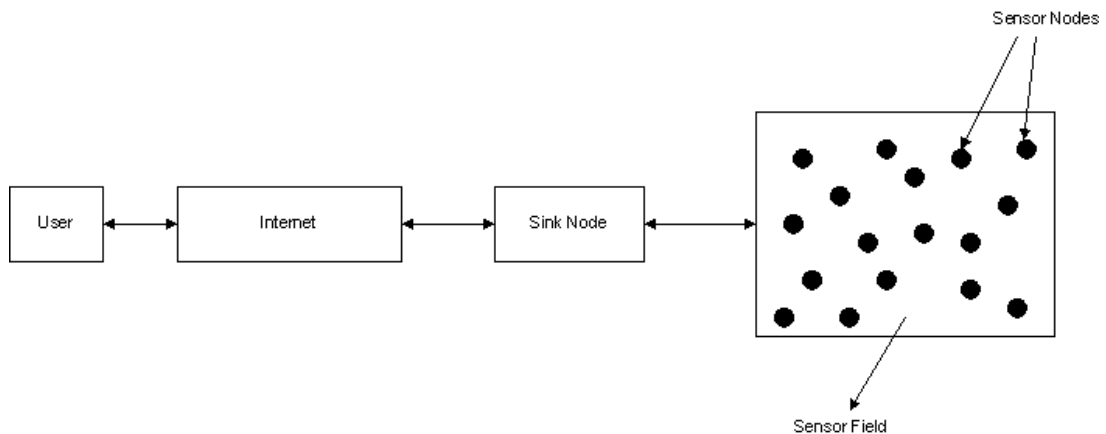


Figura 4 – Arquitetura de uma Rede de Sensores sem Fio

observadores. Esse pode ser implementado como um nó sensor, o qual será chamado de nó sorvedouro (ou *sink node*), ou por uma estação base (ou *base station*) [8]. Cabe mencionar que em uma WSN os nós sensores individualmente possuem pouca capacidade computacional e de energia, mas um esforço colaborativo entre esses permite a realização de tarefas que sozinho cada nó não poderia concretizar [140].

A literatura cita inúmeros cenários de uso de tal tecnologia [8]. No âmbito militar, as redes de sensores sem fio podem ser empregadas no monitoramento das forças amigas, na vigilância do campo de batalha, no reconhecimento das forças inimigas e do terreno, nos sistemas de orientação de disparo de armamentos, no inventário dos danos sofridos em uma batalha e na detecção e no reconhecimento de ataques nucleares, biológicos e químicos. No âmbito do controle ambiental, essas podem ser empregadas na detecção de incêndios em florestas, no mapeamento do ambiente e da biodiversidade e na detecção de inundações e de terremotos. E finalmente, no âmbito da medicina, uma WSN pode ser empregada na telemonitoração das condições fisiológicas dos pacientes, na monitoria dos médicos e dos pacientes dentro de um hospital assim como na monitoria da administração de fármacos.

Redes de sensores sem fio distinguem-se das redes de computadores convencionais em vários aspectos [8]. Normalmente, essas redes possuem um grande número de nós distribuídos, têm restrições energéticas, e devem possuir mecanismos que permitam a manifestação das pro-

priedades de auto-configuração e de adaptação, uma vez que problemas tais como falhas de comunicação e perda de nós podem ocorrer em tal tipologia de rede, já que os nós sensores interagem com o ambiente. Portanto, uma WSN tende a ser autônoma e a requerer alto grau de cooperação para que tal interação possa ocorrer.

Uma rede de sensores sem fio também se diferencia de uma rede móvel *ad hoc* [9]. Em primeiro lugar, não é possível construir um esquema de endereçamento global devido ao número relativamente grande de nós sensores. Em segundo lugar, a maioria das aplicações focadas em redes de sensores requer a transmissão dos dados coletados de múltiplas origens para um ponto de acesso particular. Em terceiro lugar, os nós sensores possuem capacidade limitada de energia, processamento e armazenamento. Em quarto lugar, na maioria das aplicações, os nós da WSN permanecem no mesmo lugar após a distribuição inicial desses, com exceção de alguns nós móveis. Em quinto lugar, as redes de sensores são geralmente específicas a uma aplicação, de modo que os requisitos da rede mudam de acordo com modificações dos requisitos daquela. Em sexto lugar, o conhecimento da posição geográfica do nó sensor é importante, pois que a coleta dos dados é normalmente baseada na localização dos nós. E finalmente, os dados coletados por diversos nós de uma WSN são tipicamente baseados no mesmo fenômeno, portanto existe uma alta probabilidade de que os dados possam apresentar certo grau de redundância.

### 2.3 Redes de Sensores e Atores sem Fio

A extensão das redes de sensores sem fio com atuadores é um campo de pesquisa mais recente, o qual permite o aperfeiçoamento de suas capacidades bem como o aumento do espectro de sua aplicabilidade a um custo que consiste na implementação de um mecanismo de *feedback* que pode trazer um certo grau de instabilidade ao sistema resultante [76]. Uma rede composta por tais dispositivos é denominada de redes de sensores e atores sem fio [7]. Da mesma forma que uma WSN, uma WSAN é um tipo especial de rede móvel *ad hoc*.

Uma rede de sensores e atores sem fio é um sistema de controle distribuído constituído por dispositivos heterogêneos referenciados como sensores e atores, os quais estão interconectados por meio de uma rede sem fio [108]. Os sensores capturam informações a respeito do ambiente e as transmitem para os atores por meio do uso de caminhos de comunicação *single-hop* ou *multi-hop*. Atores são dispositivos que possuem maior tempo de vida de bateria e maior capacidade de computação e comunicação, os quais tomam decisões a partir da informação recebida e então executam ações por meio de um ou diversos atuadores com o objetivo de alterar o comportamento do ambiente (a arquitetura de um nó ator pode ser visualizada na Figura 5 - adaptado de [7]).

Da mesma forma que em um cenário baseado em WSN, em uma WSAN os nós sensores e nós atores podem ser distribuídos de forma determinística ou randômica no campo de sensoriamento, enquanto que o nó escoadouro monitora a rede em sua totalidade e se comunica com

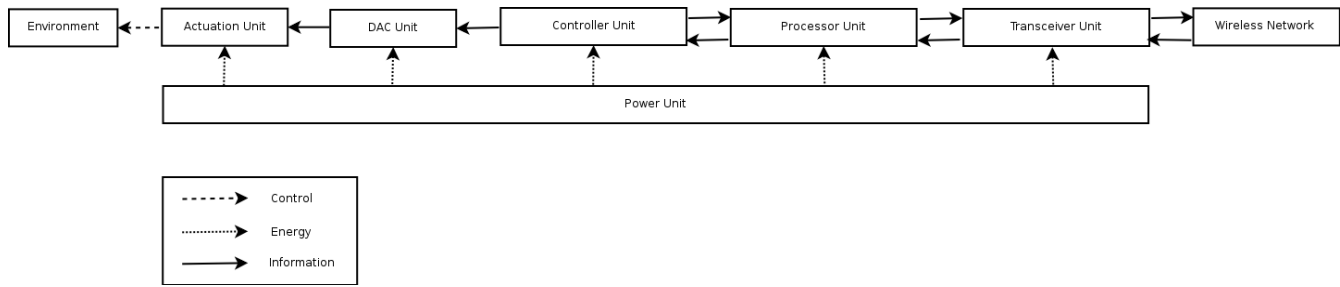


Figura 5 – Arquitetura de um Nó Ator

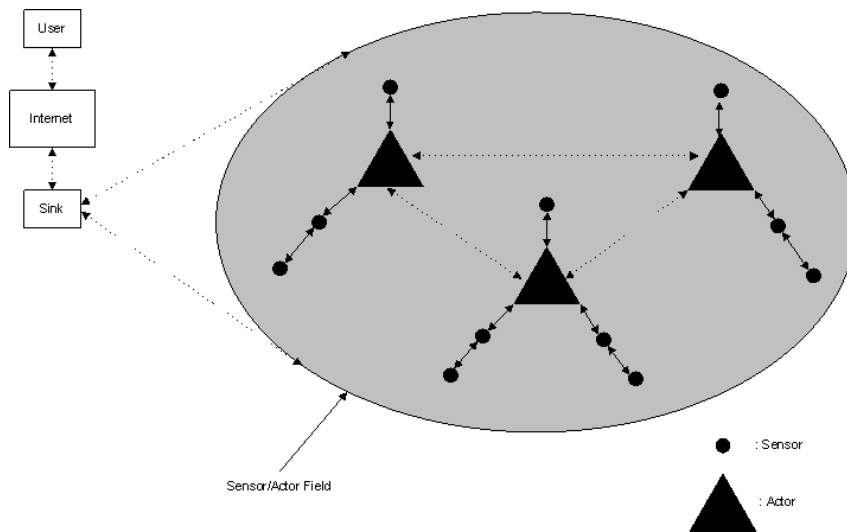


Figura 6 – Arquitetura de uma Rede de Sensores e Atores sem Fio

os nós sensores e com os nós atores (ver Figura 6 - extraído de [7]). Similarmente ao caso das redes WSN, as redes WSAN permitem o vislumbre de novas possibilidades de aplicações de tal tecnologia, tais como monitoria ambiental, automação doméstica, automação industrial e automação predial. Contudo, devido a coexistência de nós sensores e atores, constata-se que existem algumas características que diferenciam uma WSAN de uma WSN [183]. Os nós sensores e atores possuem capacidades distintas, o número de nós atores é inferior ao número de nós sensores, os nós devem ser coordenar para o sensoriamento e a execução de ações e uma WSAN possibilita a interação remota com os nós atores da rede.

Os nós sensores que detectam um evento proveniente de algum fenômeno físico podem transmitir a informação sensorizada para os nós atores, os quais podem processá-la e iniciar uma ação apropriada no ambiente, ou encaminham-na em direção ao nó escoadouro, o qual pode enviar comandos de ação para os nós atores. O primeiro tipo de organização é denominada de arquitetura automatizada (ver Figura 7 - extraído de [7]), devido ao fato de que não existe na rede um elemento que desempenhe o papel de coordenador central. O segundo tipo é denominada de arquitetura semi-automatizada (ver Figura 8 - extraído de [7]), uma vez que o nó escoadouro coleciona a informação sensorizada do ambiente e realiza a coordenação de todo o processo de alocação de tarefas aos nós atores. Dependendo do tipo de aplicação a ser pro-

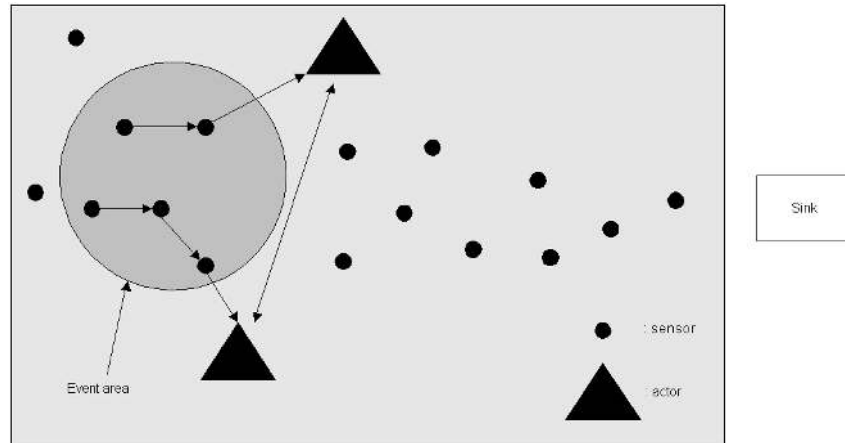


Figura 7 – Arquitetura de uma Rede de Sensores e Atores sem Fio Automatizada

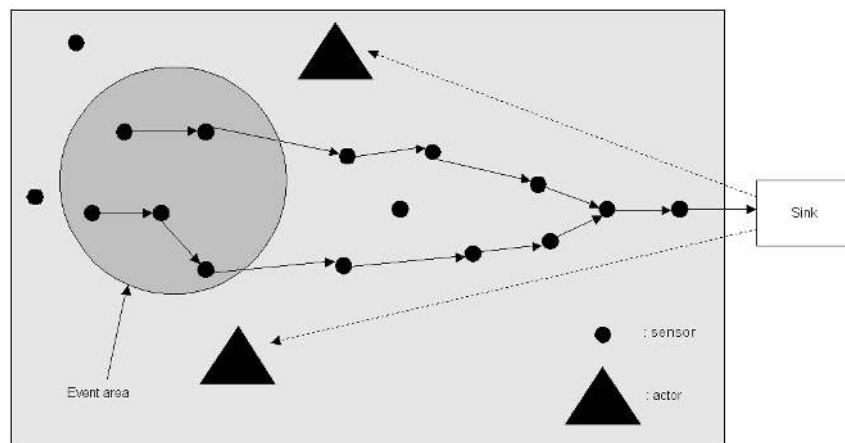


Figura 8 – Arquitetura de uma Rede de Sensores e Atores sem Fio Semi-Automatizada

jetada para uma WSN, uma dessas arquiteturas pode ser usada. A vantagem da arquitetura semi-automatizada é que ela é similar a arquitetura já empregada em aplicações desenvolvidas para as redes de sensores sem fio. Portanto, parece ser sensato afirmar que não se faz necessário o desenvolvimento de novos algoritmos e de novos protocolos para a realização das tarefas de comunicação e de coordenação em tal subtipo de WSN [7].

Outra denominação frequentemente encontrada na literatura para se referenciar a uma rede de sensores e atores sem fio é o termo redes de sensores e atores (ou *Sensor Actor Networks* - SANET). Essa é comumente composta por um conjunto de nós sensores que são estáticos (chamados *motes*) e por um conjunto de nós atores que possuem mobilidade (usualmente, robôs) [98]. Logo, a principal diferença entre ambas as tipologias reside na arquitetura do nó ator, a qual é constituída simultaneamente por dispositivos de sensoriamento e de atuação. Essa pode ser visualizada na Figura 9 (extraído de [138]).

Visto que a arquitetura de uma WSN é composta por nós que possuem tanto dispositivos de sensoriamento quanto dispositivos de atuação, ela explora por definição a metodologia



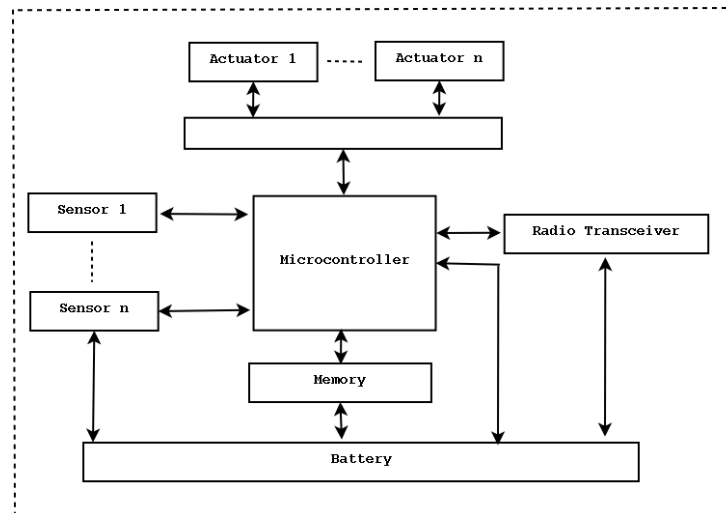


Figura 9 – Arquitetura de um Nó Ator em uma Rede do Tipo SANET

de *feedback* [181] (ou retroação), a qual é reconhecida como o elemento central da Teoria de Controle [114]. Tal mecanismo é ubíquo tanto na ciência quanto na natureza [113], e apresenta diversas vantagens no que se refere ao projeto de sistemas de controle [17]. Esse conceito possibilita a criação e a manutenção de comportamento linear a partir de componentes que se comportam de forma não linear, o que permite a um sistema de controle manifestar insensibilidade a distúrbios externos e a variações no comportamento de seus componentes individuais [14].

O campo de investigação da Teoria de Controle é basicamente multidisciplinar. Ela fornece os princípios e os métodos usados no projeto de sistemas de engenharia que conseguem manter o desempenho desejado por meio da adaptação automática a mudanças no ambiente externo [14] [113]. Mais especificamente, a Teoria de Controle refere-se ao *framework* matemático empregado na análise e na síntese de sistemas de controle, nos quais emprega-se a técnica de retroação para lidar com as incertezas derivadas da interação com o ambiente físico.

Tal tipologia de sistema constitui-se numa importante subclasse dos sistemas embarcados [11]. Um sistema embarcado é um sistema computacional cuja função primária não reside no processamento de informações, mas sim na interação permanente com o ambiente [33]. Dentre esses, destacam-se os sistemas reativos. Um sistema reativo é um sistema que reage continuamente por meio do envio de respostas a estímulos provenientes do ambiente circundante. Por sua vez, um sistema de tempo real é um sistema computacional que deve reagir a estímulos oriundos do seu ambiente em prazos específicos [63]. Esse também pode ser caracterizado como qualquer atividade de processamento de informação que deve responder a estímulos gerados externamente dentro de um período de tempo finito e especificado [182]. Em outra definição, um sistema de tempo real é um tipo de sistema que deve reagir a estímulos do ambiente dentro de intervalos de tempo ditados por esse [132]. Usualmente, constata-se na literatura que tal categoria de sistema embarcado é composto pelos seguintes elementos: um sistema a controlar, uma interface de instrumentação com o ambiente (composta de sensores e

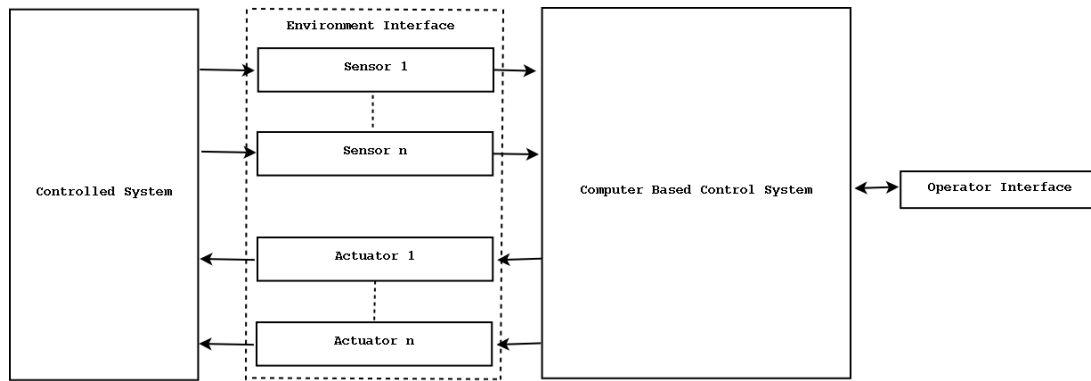


Figura 10 – Arquitetura de um Sistema de Tempo Real

atuadores), um sistema de controle baseado em computador e uma interface de operação (ver Figura 10 - adaptado de [17] [33]). Conseqüentemente, pode-se sugerir que uma WSAAN pode ser visualizada como um sistema de tempo real.

Como decorrência dessa visão conceitual, verifica-se que a corretude dessa classe de sistema não depende somente dos resultados lógicos produzidos pelo dispositivo de computação, mas também do período de tempo no qual tais resultados foram produzidos [161]. Logo, um sistema de tempo real deve manifestar a usual propriedade de corretude lógica. Em acréscimo a essa, um sistema de tempo real deve manifestar a propriedade de corretude temporal, isto é, uma aplicação de tempo real deve respeitar restrições temporais (ou *deadlines*) para que o sistema apresente o comportamento temporal desejado. Derivado do fato de que uma aplicação pode ser concluída após o seu *deadline*, define-se a seguinte classificação para sistemas de tempo real [33]:

- *hard real-time system*: nessa tipologia, é absolutamente imperativo que o sistema responda aos eventos produzidos pelo ambiente dentro do *deadline* especificado.
- *soft real-time system*: para um sistema de tempo real dessa categoria, os tempos de resposta aos estímulos externos ainda são importantes; todavia, o sistema ainda consegue manifestar a propriedade de corretude temporal se *deadlines* não forem ocasionalmente atendidos.

Conforme mencionado anteriormente, uma rede de sensores e atores sem fio deve prover o atendimento de limitações temporais, o que pode depender do cenário particular de aplicação de tal tecnologia. Como decorrência, o desenvolvimento de mecanismos de coordenação poderá ser influenciado por tal diretiva, de modo tal que o sistema resultante venha a manifestar um comportamento temporal que possa ser expresso por uma das classificações acima explicitadas.

### 3 Mecanismos de Coordenação Ator-Ator

Como decorrência do crescente uso de técnicas de computação distribuída resulta em uma comunidade de processos concorrentes, ou ecossistema computacional, o qual em suas interações, estratégias e falta de conhecimento perfeito é análogo aos ecossistemas biológicos e às economias humanas [82]. Visto que um sistema distribuído consiste de um número relativamente grande de elementos independentes que competem por recursos para a resolução de problemas ou que interagem diretamente com o ambiente, essa analogia permite que se introduzam novas perspectivas a respeito de como se projetar e de como se compreender o comportamento desses sistemas computacionais [81]. De particular interesse é o fato de que tais sistemas apresentam propriedades que lhes permitem operar de forma coordenada, mesmo que sem conhecimento perfeito, as quais lhes possibilitam adaptar-se a mudanças no ambiente ou a perturbações em seus componentes individuais.

Uma vez que uma WSAAN é um sistema cuja função primária é a interação permanente com o ambiente, parece razoável pressupor que essa visão de computação enquanto ecossistema pode ser aplicada na elaboração de protocolos baseados em metáforas econômicas e em metáforas biológicas para a coordenação ator-ator. Tais metáforas permitem a um sistema computacional complexo adequar-se a condições ambientais mutáveis [35]. Ademais, em redes *ad hoc* é usual o emprego de esquemas localizados em oposição ao que ocorre em alguns protocolos existentes, os quais requerem informação sobre o estado global do sistema [163]. Protocolos localizados produzem soluções escaláveis [108]. Esses possibilitam o desenvolvimento de algoritmos para redes *ad hoc* que possuem um número arbitrário de nós por meio do uso de informação local, a qual é obtida a partir das propriedades dos nós vizinhos [180]. Mais especificamente, esse conhecimento localizado é derivado da relação de vizinhança que se estabelece entre um nó e os outros nós que estão contidos no alcance de transmissão do seu dispositivo de rádio. Por conseguinte, os algoritmos sugeridos a seguir para a coordenação ator-ator exploram somente informação local, os quais eventualmente poderão vir a manifestar a propriedade de escalabilidade.

É interessante observar que tal diretiva de projeto, onipresente no contexto dos sistemas distribuídos enquanto categoria genérica, remonta as suas origens do neo-classicismo econômico. Em 1945, o economista Friedrich Hayek [77] argumentou que o sucesso de um sistema econômico baseado em mecanismos de mercado se deve, em quase a sua totalidade, à maneira distribuída pela qual esses mecanismos operam, e que esses funcionam de forma mais efetiva quando os indivíduos atuam de forma autônoma, com base somente em informação local.

Visto que os mecanismos de coordenação ator-ator devem em sua essência viabilizar a alo-

cação dinâmica de tarefas entre os atores, verifica-se na literatura direcionada para o âmbito dos sistemas embarcados uma certa proeminência de abordagens baseadas nessa visão de sistema distribuído enquanto ecossistema computacional [3] [72] [74] [47] [62]. Tais abordagens são divididas em duas categorias principais: baseadas no comportamento dos mecanismos de mercado e baseadas no comportamento coletivo dos insetos. A primeira apoia-se na negociação entre os elementos envolvidos, os quais se envolvem nesse processo com o objetivo de cooperar no desempenho de uma dada tarefa. Tal paradigma requer a existência de algum mecanismo que permita aos elementos envolvidos se comunicarem, sem o qual a ação cooperativa não pode ser garantida. No entanto, esse requisito de comunicação implica na limitação da escalabilidade do sistema [74]. A segunda inspira-se em metáforas biológicas, e se baseia no comportamento coletivo das sociedades de insetos. Nesse modelo, cada elemento do sistema possui um estímulo associado a cada tarefa que deve ser executada, o qual depende de parâmetros intrínsecos a cada elemento. Quando o nível de um dado estímulo excede um determinado limiar, o elemento inicia o desempenho de uma dada tarefa. Esse tipo de sistema não requer que existam mecanismos de comunicação entre os seus elementos. Portanto, ele é mais escalável do que um sistema baseado em negociação [74].

Em uma WSN tarefas podem ser atribuídas a somente um ator (cenário de uma tarefa atribuída a um único ator, ou *Single-Actor Task* - SAT) ou a um conjunto de atores (cenário de uma tarefa atribuída a múltiplos atuadores, ou *Multi-Actor Task* - MAT). De acordo com a literatura [7], o caso SAT consiste na seleção do nó ator que possui um raio de atuação que é capaz de cobrir toda a área de eventos. Já o caso MAT consiste na seleção do subconjunto de atores cuja a união dos raios de atuação garante a satisfação dessa propriedade. A partir dessa classificação, o caso MAT foi escolhido como alvo dos mecanismos de coordenação propostos, pois que o caso SAT é uma instância particular do caso MAT. Dessa maneira, a instância do problema da coordenação ator-ator focalizada pelo presente trabalho consiste na determinação do subconjunto mínimo de atores que devem atuar em um sistema que deve ser controlado.

Nas seções que se seguem, descrevem-se as pressuposições assumidas, o modelo de sistema adotado, os trabalhos relacionados bem como os algoritmos propostos para a coordenação ator-ator em cada estratégia a ser investigada, seja essa baseada no comportamento dos mecanismos de mercado ou seja essa baseada no comportamento coletivo dos insetos. Tais algoritmos são totalmente distribuídos, e empregam somente informação local no processo de colaboração.

### 3.1 Notações Empregadas na Apresentação dos Mecanismos

A Tabela 1 descreve um conjunto de notações que será empregado na apresentação dos mecanismos elaborados para a coordenação ator-ator:

Tabela 1 – Notações Empregadas nos Mecanismos de Coordenação Ator-Ator Sugeridos

Variável/Constante	Descrição
$n_{a_i}$	nó ator $i$
$sc_j$	sistema controlado $j$
$d_{a_{ki}}$	dispositivo de atuação $k$ acoplado ao nó ator $i$
$ff_m$	fenômeno físico $m$
$ae_m$	área de eventos gerada pelo raio de propagação do fenômeno físico $m$
$n_{s_o}$	nó sensor $o$
$d_{s_{po}}$	dispositivo de sensoriamento $p$ acoplado ao nó sensor $o$
$c_r$	contrato $r$
$dae_m$	diâmetro da área de eventos $m$
$d_{im}$	distância euclidiana entre o ator $i$ e a área de eventos associada ao fenômeno físico $m$ , isto é, $d_{im} = \sqrt{(x_i - x_m)^2 + (y_i - y_m)^2}$
$e_i$	quantidade de energia consumida pelo ator $i$
$t_{end_{ikl}}$	tempo que o ator $i$ leva por meio do atuador $k$ para completar a execução da tarefa $l$
$r_{ik}$	raio de atuação do atuador $k$ que está acoplado ao ator $i$
$t_{idle_i}$	tempo no qual o ator $i$ esteve no estado <i>idle</i>
$\theta$	limiar de resposta
$\theta_{min}$	limiar de resposta mínimo
$\theta_{max}$	limiar de resposta máximo
$\alpha$	coeficiente que modula a influência da variável $d_{im}$
$\beta$	coeficiente que modula a influência da variável $e_i$
$\gamma$	coeficiente que modula a influência da variável $t_{end_{ikl}}$
$\lambda$	coeficiente que modula a influência da variável $r_{ik}$
$\rho$	coeficiente que modula a influência da variável $t_{idle_i}$
$\omega$	coeficiente que modula a influência da variável $\theta$
$\delta$	limite temporal para a finalização de uma ação no ambiente

## 3.2 Pressuposições

Na literatura da área de Sistemas Distribuídos é usual que um conjunto de pré-condições seja assumido, pois somente dessa maneira os problemas definidos em cada proposta de solução podem ser resolvidos computacionalmente [167] [51] [103] [152]. Conseqüentemente, a seguir são explicitadas as pressuposições subjacentes aos mecanismos de coordenação ator-ator propostos no presente capítulo:

- Arquitetura da rede: assume-se que a WSAAN possui uma arquitetura automatizada. Logo, assume-se que não existe uma entidade central que desempenha as funções de coordenação e de coleta dos dados.
- Localização: assume-se que tanto os nós sensores quanto os nós atores podem determinar a sua localização por meio do emprego de algoritmos de localização [31] ou por meio do emprego da tecnologia GPS (*Global Positioning System*) [107]. Além disso, assume-se que os dispositivos de sensoriamento acoplados aos nós sensores podem determinar a posição geográfica e o raio de propagação do fenômeno físico que difundiu algum sinal para o ambiente. Essas pressuposições parecem ser razoáveis, uma vez que os nós de uma WSAAN devem poder identificar a localização dos eventos, o que é pré-condição para a

monitoria e o controle do ambiente [116].

- Sincronização de tempo: assume-se que a rede é sincronizada por meio de algum protocolo de sincronização de tempo existente [164].
- Roteamento: assume-se que os pacotes da rede são roteados por meio de algum protocolo de roteamento existente [2].
- Controle de topologia: assume-se que a conectividade entre os nós atores da rede é garantida por meio de algum protocolo existente [1] [148] [5].
- Tempo real: assume-se que as aplicações que estão sendo executadas nos diversos nós da WSAN estão submetidas ao modelo *soft real-time system*.
- Raio de atuação: assume-se que todos os dispositivos de atuação possuem o mesmo raio de atuação.
- Consumo de energia: assume-se que a energia requerida dos atores para atuar no ambiente controlado é maior do que a energia requerida para a comunicação [108]; por sua vez, assume-se que a energia requerida para a comunicação é maior do que a energia requerida para a computação [79].
- Interação com o ambiente: assume-se que  $\forall j \exists m : (sc_j \subset ae_m)$ .

### 3.3 Modelo de Sistema

No modelo de sistema adotado, são abstraídas as especificações tecnológicas dos dispositivos que possam viabilizar a implementação concreta de uma rede de sensores e atores sem fio. O sistema é modelado como um conjunto de nós que possuem capacidade computacional e de comunicação via rede sem fio. Um nó sensor é equipado com um conjunto de dispositivos de sensoriamento, o qual é caracterizado principalmente pela quantidade de energia disponível e pelo custo energético para a computação, para o sensoriamento e para a comunicação de dados. Por outro lado, um nó ator é equipado com um conjunto de dispositivos de atuação, o qual caracteriza-se por uma maior capacidade computacional, de comunicação e energética do que os nós sensores. O dispositivo de comunicação de cada nó realiza a difusão de mensagens para os dispositivos vizinhos dentro de um intervalo  $x$ , o qual pode variar a partir da configuração da potência de transmissão do dispositivo para um nível apropriado. Nesse contexto, um nó  $y$  é considerado vizinho de um nó  $z$  se a distância entre ambos for inferior a  $x$ . O dispositivo de sensoriamento captura eventos do ambiente que estão dentro do seu intervalo de leitura mínimo e máximo, enquanto que o dispositivo de atuação efetua a ação de controle sobre um sistema controlado desde que aquele possa se conectar a esse por meio de uma interface de acoplamento

que esteja dentro do seu raio de atuação. Além dessas duas entidades, incluem-se a esse modelo um conjunto de fenômenos físicos e um conjunto de sistemas controlados. O primeiro tipo de entidade propaga sinais para o mundo externo, enquanto que o segundo tipo permite o controle efetivo do ambiente por meio da ação de um dispositivo de atuação que possa ser conectado a alguma de suas interfaces de acoplamento.

Particularmente, o presente modelo de sistema assume que:

- Um sistema controlado  $sc_j$  deve estar contido no interior de uma área de eventos  $ae_m$ .
- Todo nó ator  $n_{a_i}$  deve possuir a capacidade de se acoplar ao sistema controlado  $sc_j$  para que a ação de controle seja possível. Portanto, deve existir um ponto de intersecção entre cada ator  $n_{a_i}$  e o sistema controlado  $sc_j$ , assim como o dispositivo de atuação  $d_{a_{ki}}$  de cada ator  $n_{a_i}$  deve ser compatível com alguma interface de acoplamento do sistema controlado  $sc_j$ .
- Um nó ator  $n_{a_i}$  possui um dispositivo de atuação  $d_{a_{ki}}$ , o qual possui um raio de atuação (o qual é definido como a extensão de um braço mecânico que pode ser empregado para a execução de ações no ambiente).
- Um sistema controlado  $sc_j$  possui um raio de controle, o qual define a área de influência de  $sc_j$ .
- Um fenômeno físico  $ff_m$  possui um raio de propagação, o qual define a área de eventos gerada por  $ff_m$ .

### 3.4 Baseados em Abordagens Econômicas

Mecanismos de mercado desenvolveram-se gradualmente no âmbito das sociedades humanas, e constituem-se em um sistema de decisão distribuído, robusto e altamente efetivo, cujo propósito fundamental reside na alocação de recursos e na coordenação da atividade econômica [84]. Tais mecanismos são algumas vezes usados para a identificação de qualquer sistema ativo no qual os preços e as decisões dos agentes econômicos são mutualmente dependentes, sendo que a alocação de recursos é determinada pela relação de equilíbrio dinâmico que se estabelece entre esses.

Existem várias definições para o conceito de mercado. Uma dessas afirma que o mercado consiste em qualquer contexto no qual ocorre a compra e a venda de bens e de serviços [124]. Outra definição afirma que um mercado consiste em um meio que possibilita a ocorrência de trocas entre compradores e vendedores [142]. Convencionalmente, assume-se que os participantes do mercado são racionais, isto é, que eles procuram maximizar a sua utilidade e que eles atuam de forma consistente na perseguição desse objetivo. Dessa pressuposição, surge um

critério mínimo para a definição de um sistema enquanto mercado: um mercado é um meio no qual agentes econômicos autônomos trocam mercadorias de acordo com seus preços tendo em vista a maximização de suas próprias utilidades [126].

Na literatura microeconômica, pode-se constatar a existência de mecanismos que permitem o funcionamento do mercado [173], os quais possibilitaram o êxito econômico obtido pela maioria das sociedades modernas. Dentre esses, os que têm maior relevância no contexto dos sistemas distribuídos são, a saber: a teoria do equilíbrio geral e os leilões. Tais mecanismos são descritos de forma mais detalhada a seguir:

- Na teoria do equilíbrio geral, dada uma economia que consiste de múltiplos mercados interconectados, o equilíbrio geral (ou equilíbrio de mercado, ou ainda, equilíbrio walrasiano <sup>1</sup>) é o estado global no qual a oferta e a demanda se igualam simultaneamente em todos os mercados, dado que todos os agentes econômicos visam à maximização do seu grau de satisfação individual. Dito de outra maneira, a quantidade total que cada indivíduo deseja comprar de cada bem aos preços correntes é igual a quantidade total disponível.
- Os leilões são mecanismos pelos quais se identificam as preferências dos compradores com o objetivo de se atingir a propriedade do preço de equilíbrio, dado pela interação entre a oferta e a procura. Existem várias modalidades de leilões, sendo que as mais conhecidas são: o leilão inglês, o leilão holandês, o leilão de lance fechado e o leilão de Vickrey. No leilão inglês, o leiloeiro parte de um preço de reserva, que é o menor preço pelo qual o vendedor se desfará do bem. Os participantes oferecem, sucessivamente, preços mais altos, sendo que em geral cada lance excede o anterior por algum incremento mínimo. Quando nenhum participante se dispõe a um novo lance, o bem é vendido àquele que apresentou o lance mais alto. No leilão holandês, ocorre justamente o contrário. O leiloeiro inicia com um preço alto e o reduz gradualmente até que algum comprador o adquira. No leilão de lance fechado, cada participante escreve seu lance em um pedaço de papel e o coloca em um envelope fechado. Esses são reunidos e abertos e o bem cabe a quem apresentou o maior lance, que paga pelo bem o montante registrado em seu lance. Finalmente, o leilão de Vickrey é uma variação do leilão de lance fechado. A diferença reside no fato de que em uma instância de negociação do leilão de Vickrey, o bem cabe ao comprador que submeteu o lance mais elevado; contudo, o vencedor paga pelo bem o segundo preço mais elevado.

O principal ponto forte de uma economia regida por tais mecanismos reside em sua habilidade de lidar com condições mutáveis com significativo grau de sucesso [59], a respeito das quais se possui informação incompleta. Visto que tal tipologia econômica não depende de uma estrutura hierárquica para a coordenação e para a atribuição de tarefas, ela pode manifestar um nível expressivo de robustez contra mudanças ambientais e de flexibilidade para

<sup>1</sup>Leon Walras (1834-1910) foi um dos primeiros pesquisadores da teoria de do equilíbrio geral.



se adaptar a condições dinâmicas. Portanto, um modelo de computação distribuída baseado em mecanismos de mercado permite que seus elementos possam interagir com ambientes dinâmicos de uma maneira adaptativa e oportunística, de modo tal que esses possam responder rapidamente a mudanças nas condições ambientais. Nessa direção, argumentos apresentados em [109] [83] sugerem que mercados automatizados trazem muitas vantagens para o projeto e para a implementação de sistemas distribuídos. Tanto que a adoção de tais metáforas econômicas no contexto dos sistemas distribuídos não é recente [67] [176] [69] [178] [177], visto que algoritmos baseados na analogia de mercado permitem o controle de sistemas complexos que estão inseridos ambientes que possuem relativos graus de incerteza e de imprevisibilidade [35].

Tanto a teoria do equilíbrio geral quanto os mecanismos de leilão se baseiam em transações que ocorrem entre compradores e vendedores através interações que ocorrem no mercado. Tais transações são conduzidas por meio do emprego de estratégias de negociação, as quais podem ser usadas para resolução de conflitos e para a coordenação de atividades em um variedade de problemas no âmbito computacional, dentre os quais, a atribuição de tarefas [112] [95]. Esse problema em peculiar consiste na essência da questão da coordenação ator-ator em uma WSA. Portanto, a partir de um processo de negociação, um componente de um sistema distribuído pode tentar contratar outro componente para a execução de uma tarefa que aquele sozinho não consegue desempenhar. O que significa na esfera de uma WSA que por meio de um processo de negociação, um ator poderia tentar contratar outro ator ou um conjunto de atores para desempenhar uma tarefa que deve ser executada no ambiente controlado.

A negociação entre agentes inteligentes é uma das questões fundamentais de pesquisa no âmbito dos sistemas multi-agentes [6]. Nesse contexto, um dos mais estudados protocolos para a negociação entre sistemas computacionais encontra suas origens na metáfora das organizações humanas [54]. O protocolo de redes de contratos (ou *Contract-Net Protocol* - CNP) [156] é um *framework* genérico de negociação que tem se mostrado uma das mais influentes abordagens empregadas no projeto de sistemas multi-agentes [85]. O CNP é baseado nos processos de contratação que ocorrem nos mais diversos tipos de organizações presentes nas sociedades humanas, e é uma variação dos mecanismos de negociação distribuídos baseados no conceito de equilíbrio geral para o problema da alocação de tarefas [143]. Essencialmente, o CNP é um mecanismo de coordenação composto por um conjunto de primitivas de comunicação que permite interações de alto-nível entre nós que devem cooperar para a execução de tarefas [155]. Já que o CNP é um *framework*, ele somente indica que primitivas de comunicação devem ser empregadas no processo de contratação; no entanto, o conteúdo real dos *slots*<sup>2</sup> de cada tipo de primitiva é dependente do domínio em particular. Cabe mencionar que o CNP é largamente usado no âmbito dos sistemas multi-agentes para a alocação dinâmica de tarefas entre agentes cooperativos [110].

O CNP é um protocolo de negociação localizado e descentralizado, no qual um contrato  $c_r$  consiste de um acordo explícito entre um agente que gera uma tarefa (*manager*) e um agente

<sup>2</sup>Um slot consiste em uma unidade de medida que subdivide o conteúdo de uma mensagem do CNP.

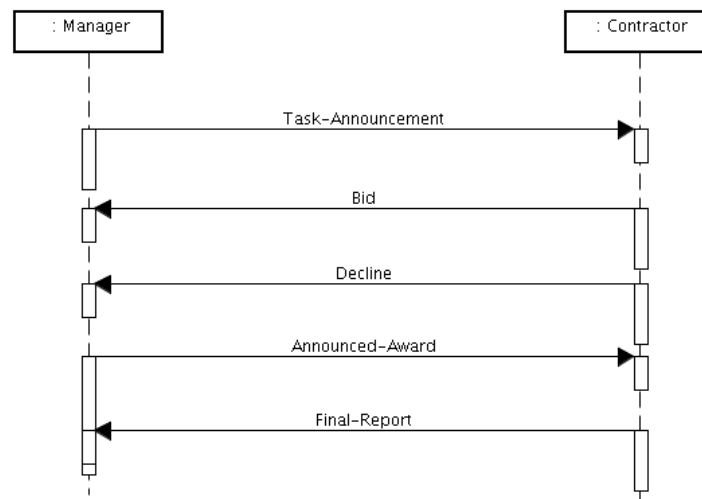


Figura 11 – Diagrama de Sequência em UML da Versão Base do Protocolo de Redes de Contratos

que deseja executar essa tarefa (*contractor*) [154]. Por conseguinte, tais agentes coordenam suas atividades por meio de contratos, os quais são dinamicamente criados para que finalidades específicas sejam atingidas. Particularmente, o *manager* é responsável por monitorar a execução de uma tarefa e por processar os resultados dessa execução, enquanto que o *contractor* é responsável pela efetiva execução da tarefa. Tais papéis não são designados *a priori*; esses são determinados dinamicamente, a partir da demanda corrente de processamento do sistema [156]. De acordo com [155], constata-se que existe uma versão base do CNP bem como variações dessa que permitem a introdução de extensões ao CNP.

Argumenta-se que um processo de negociação pode ser introduzido em um mecanismo de coordenação ator-ator em redes de sensores e atores, o qual consiste em uma instância particular do problema da alocação dinâmica de tarefas [7]. Problemas de alocação dinâmica de tarefas podem ser resolvidos de forma descentralizada pelo CNP [54]. Portanto, nessa seção são propostos algoritmos que se baseiam no reuso de tal abordagem.

### 3.4.1 O Protocolo de Redes de Contratos

Na versão base do CNP, um contrato é estabelecido por um processo de seleção mútua, o qual se baseia em um mecanismo de troca de mensagens bidirecional. Tais mensagens podem ser visualizadas no diagrama de sequência em UML [28] contido na Figura 11.

Em uma instanciação do processo de negociação da versão base do CNP, o *manager* de uma dada tarefa elabora um contrato  $c_r$ , o qual possui um *deadline* para o recebimento de ofertas, e notifica a sua existência aos potenciais contratantes por meio do envio em modo *broadcast* de uma mensagem *task announcement*. Os agentes disponíveis (*contractors*) para a execução de tarefas avaliam as mensagens *task announcement* recebidas, e submetem lances por meio

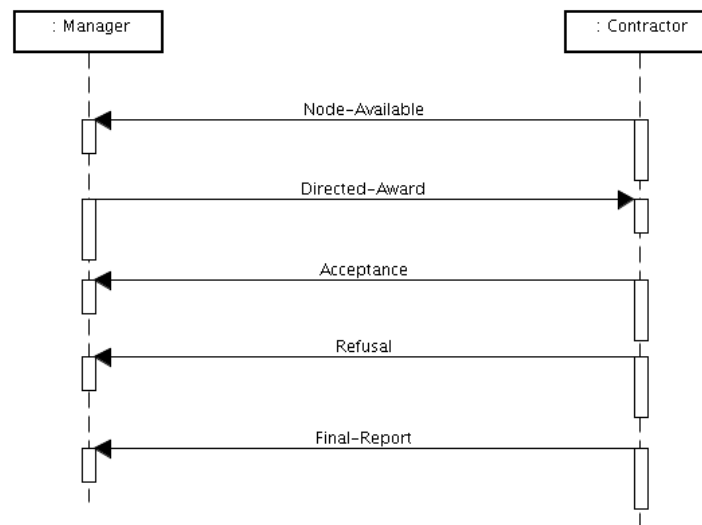


Figura 12 – Diagrama de Sequência em UML da Versão Estendida do Protocolo de Redes de Contratos

do envio de mensagens *bid* para as tarefas para as quais eles são qualificados. Um contratante somente submete um lance quando esse atende à especificação de elegibilidade de um contrato, a qual define um conjunto de critérios que devem ser atendidos para a submissão de ofertas. Após a expiração do *deadline* associado ao contrato  $c_r$ , o *manager* conduz a avaliação dos lances submetidos e concede contratos por meio do envio de mensagens *announced award* para os contratantes que são considerados os mais aptos. Naturalmente, um contratante pode se recusar a executar uma tarefa, e pode notificar o *manager* de tal decisão a partir do envio de uma mensagem *decline*. Ao término da execução de uma tarefa, é facultativo ao contratante a notificação do gerente via envio da mensagem *final report*.

A partir da constatação de que esse mecanismo básico de negociação apresenta limitações, foram propostas extensões ao CNP com o objetivo de mitigá-las [155]. De peculiar interesse ao presente trabalho é o conceito de contrato dirigido. Esse possibilita a simplificação desse mecanismo de negociação, o que pode implicar no aumento da eficiência do protocolo devido a diminuição do custo de comunicação entre os agentes envolvidos em uma instanciação de um contrato  $c_r$ . As mensagens envolvidas nessa modalidade abreviada do CNP podem ser visualizadas no diagrama de sequência em UML contido na Figura 12.

No contexto dessa extensão particular, se o *manager* sabe exatamente quais contratantes são os mais apropriados para o desempenho de uma tarefa particular associada a um contrato  $c_r$ , uma vez que mensagens *node available* tenham sido previamente recebidas e armazenados em uma lista, um contrato dirigido pode ser concedido. Cada mensagem *node available* possui um *deadline* para a recepção de mensagens do tipo *directed award*, a qual lhe delega a execução da tarefa correspondente ao contrato  $c_r$ . Se uma dada instância de uma mensagem *node available* está expirada, ela não é empregada no mecanismo de determinação dos contratantes mais aptos. Portanto, nessa versão do CNP não se faz necessário o envio de mensagens *task announcement*

para a notificação da existência de tarefas a executar, de maneira tal que lances não necessitam ser submetidos. Ao invés disso, uma mensagem *directed award* é enviada ao contratante ou contratantes previamente selecionados, sem que se faça necessário a execução de uma rodada do processo de negociação. Se um contratante que recebe uma mensagem *directed award* não preenche a especificação de elegibilidade de um contrato, ele pode enviar uma mensagem *refusal* para o gerente. Em caso contrário, ele envia-lhe uma mensagem *acceptance*. Da mesma forma que no CNP base, é facultativo o envio da mensagem *final-report*.

### 3.4.2 Baseado no Protocolo de Redes de Contratos

No âmbito da coordenação ator-ator, pressupõe-se que o ator mais apto é aquele que está mais próximo da área de eventos, que possui a maior quantidade de energia residual, que poderá levar o menor tempo para completar a execução da ação e que possui o maior raio de atuação [7]. A partir dessas pressuposições, foi definida a função de custo a seguir explicitada.

A computação do grau de aptidão de um nó ator  $i$  para desempenhar uma tarefa  $l$  é dado pela função de custo definida pela Equação 3.1, enquanto que um lance submetido por  $i$  é definido pela Equação 3.2. Nesse modelo, quanto maior for o valor computado para a função de custo  $fc_i$  associada a um ator  $i$ , menor é a probabilidade de que esse venha a ser escolhido pelo nó responsável pela alocação da tarefa  $l$  em uma dada rodada do mecanismo de coordenação corrente.

$$fc_i = \frac{\alpha d_{im} + \beta e_i + \gamma t_{end_{ikl}}}{\alpha + \beta + \gamma} + \frac{1}{\lambda r_{ik}} \quad (3.1)$$

$$Bid_i = \frac{1}{fc_i} \quad (3.2)$$

A interpretação dos termos da função de custo definida pela Equação 3.1 é discutida a seguir. O primeiro termo calcula a média ponderada das variáveis  $d_{im}$ ,  $e_i$  e  $t_{end_{ikl}}$ . Logo, quanto menor for a distância entre o nó ator  $i$  e a área de eventos  $m$ , quanto menor for a quantidade de energia dispendida pelo nó  $i$  até o momento da submissão da oferta e quanto menor for o tempo que leva o ator  $i$  via atuador  $k$  para completar a execução da tarefa  $l$ , menor será a contribuição desse termo para o valor total da função de custo, e portanto, maior será o lance submetido pelo ator  $i$ . Complementarmente, quanto maior for o raio de atuação do atuador  $k$  conectado ao ator  $i$ , menor será o valor calculado para o segundo termo, e por conseguinte, maior será o lance.

A adaptação da versão base do CNP ao problema da coordenação ator-ator é discutida a seguir. Após a recepção de dados sensoriados por um nó sensor  $n_{s_o}$ , o nó ator receptor  $n_{a_i}$  assume o papel de gerente do processo de negociação. Inicialmente, esse verifica se a tarefa a ser desempenhada é do tipo SAT ou MAT. Isso se dá por meio de um procedimento que computa o número de contratantes necessários em uma dada rodada do protocolo a partir da comparação

do raio de propagação da área de eventos com o raio de atuação do gerente. A partir da execução desse procedimento, três ramificações são possíveis:

- Se for do tipo SAT, o próprio gerente tenta iniciar a ação imediatamente, desde que a proposição  $\exists i, j, k, m : ((sc_j \subset ae_m) \wedge (d_{a_{ki}} \cap sc_j))$  seja verdadeira.
- Se a proposição acima for falsa, o gerente cria uma instância de contrato associada a tarefa que deve ser executada pelo nós atores e inicia o processo de negociação por meio do envio de uma mensagem *task announcement* via *broadcast* para a rede, o que lhe notifica a respeito da existência de uma tarefa a ser desempenhada. Após a expiração do tempo de recebimento de lances associado a esse contrato em específico, o gerente avalia os lances recebidos por meio da computação da função de custo definida pela Equação 3.2, seleciona o ator mais apto (o qual submeteu a maior oferta) e envia uma mensagem *announced award* para esse, concedendo-lhe a execução do contrato corrente. Um contratante verifica se a proposição  $\exists i, j, k, m : ((sc_j \subset ae_m) \wedge (d_{a_{ki}} \cap sc_j))$  é verdadeira ao receber uma mensagem *task announcement*. Em caso afirmativo, submete um lance (o qual contém um conjunto de variáveis que descrevem o seu nível de aptidão) para o contrato indicado na mensagem de notificação e envia uma mensagem *bid*; do contrário, envia uma mensagem *decline*. Ao receber uma mensagem *announced award*, o contratante executa a ação no sistema controlado e ao final dessa atividade envia uma mensagem *final report*.
- Se for do tipo MAT, são executados todos os passos acima mencionados. Com exceção do fato de que na avaliação da função de custo, o gerente seleciona o subconjunto mínimo de atores que devem atuar em um dado sistema controlado. Para tal, ele ordena os lances que foram computados pela Equação 3.2 em ordem descendente e envia mensagens *announced award* para os atores selecionados até que a propriedade de cobertura da área de eventos seja satisfeita.

### 3.4.3 Baseado em Extensões ao Protocolo de Redes de Contratos

A adaptação da extensão particular do CNP com o constructo de contrato dirigido ao problema da coordenação ator-ator é discutida a seguir. Assim como no mecanismo anterior, o nó ator  $n_{a_i}$  que recebeu dados sensorizados de um dado nó sensor  $n_{s_o}$  assume o papel de gerente do processo de negociação. Da mesma forma, ele verifica a tipologia da tarefa a ser executada. Analogamente, três ramificações são novamente possíveis:

- Se for do tipo SAT, o próprio gerente inicia a ação imediatamente desde que a proposição  $\exists i, j, k, m : ((sc_j \subset ae_m) \wedge (d_{a_{ki}} \cap sc_j))$  seja verdadeira.
- Se a proposição acima for falsa, o gerente cria uma instância de contrato dirigido associado a tarefa que deve ser desempenhada pelo conjunto dos nós atores e conduz a

avaliação de mensagens *node available* previamente recebidas de potenciais contratantes. Nesse procedimento, o gerente seleciona o ator mais apto por meio da computação da função de custo definida pela Equação 3.2 e envia para esse uma mensagem *directed award*, o que lhe concede a execução do contrato dirigido. Um contratante verifica se a proposição  $\exists i, j, k, m : ((sc_j \subset ae_m) \wedge (d_{a_{ki}} \cap sc_j))$  é verdadeira ao receber uma mensagem *directed award*. Em caso afirmativo, envia uma mensagem *acceptance*, executa a ação no sistema controlado e ao final dessa atividade envia uma mensagem *final report*. Em caso negativo, envia uma mensagem *decline*.

- Se for do tipo MAT, são executados todos os passos acima apresentados. Com exceção do fato de que na avaliação da função de custo o gerente seleciona o subconjunto mínimo de atores que devem executar o contrato dirigido. Para tanto, computa-se a função de custo definida pela Equação 3.2 para cada mensagem *node available*, ordena-se o conjunto resultante em ordem decrescente e enviam-se mensagens *directed award* para os atores selecionados até que a propriedade de cobertura da área de eventos seja satisfeita.

Na extensão proposta nessa subseção, cada nó ator envia periodicamente via *broadcast*, a cada intervalo de disseminação, uma mensagem *node available*. Essa contém um conjunto de variáveis que descrevem o grau de aptidão do remetente da mensagem. Porquanto os valores desse conjunto de variáveis podem sofrer alterações devido a própria dinamicidade do sistema, cada mensagem desse tipo contém um tempo de expiração.

### 3.4.4 Trabalhos Relacionados

Embora alguns trabalhos publicados recentemente tenham endereçado aspectos da comunicação e da coordenação em redes de sensores e atores, evidências constatadas na literatura sugerem que ainda há espaço para contribuições. A seguir, discutem-se algumas propostas para a coordenação ator-ator no contexto de tais redes, as quais empregam princípios derivados da teoria econômica.

Em [71] é formulado um modelo de mercado para a alocação de tarefas baseado na teoria do equilíbrio geral para a coordenação ator-ator. Tal equilíbrio é descrito por um vetor de preços atribuído a cada mercado. Esse pode ser obtido por meio de um processo de ajustamento iterativo dos preços, o qual é conhecido na literatura pela denominação de *tatônnement* [178]. No modelo proposto, os nós da rede são economicamente racionais, e participam de uma economia baseada na contratação de tarefas. A economia é composta por múltiplos mercados, um para cada tarefa a ser contratada. Os contratantes competem por contratos por meio do emprego de um mecanismo baseado na abstração de leilão [93], e submetem lances baseados no custo esperado da execução das tarefas. Após a execução dessas, a cada contratante é paga uma quantidade de unidades monetárias como contraprestação do serviço prestado, sendo que o lucro

do contratante é dado pela diferença entre esse pagamento e o custo incorrido na execução da tarefa. No entanto, ainda não foram publicados resultados que possam provar a efetividade de tal abordagem.

Em trabalho mais recente [108], é proposto um modelo de otimização baseado na teoria microeconômica que visa à maximização da energia residual de cada ator para uma instância da coordenação ator-ator na qual o disparo do mecanismo de atuação é orientado a eventos específicos. Nessa publicação, a topologia dos nós da rede é reconfigurada a cada evento detectado, de modo que os atores se coordenam entre si para que a ação seja executada no ambiente supervisionado a cada ciclo de captura de eventos que são de interesse do sistema. Derivada desse modelo, é proposta uma solução distribuída inspirada no comportamento dos agentes que estão participando de um leilão do tipo lance fechado. Tal leilão visa à seleção do melhor conjunto de atores para executar uma dada tarefa. Portanto, nesse cenário as áreas nas quais estão sendo capturados os eventos são vistas como mercadorias que podem ser negociadas pelos atores. Em cada área de captura de eventos na qual existe mais de um ator apto a executar a ação é conduzido um leilão localizado, sendo que as ofertas são submetidas somente uma vez para que o *overhead* possa ser reduzido [105]. Pelo que fica evidenciado nessa proposta, a determinação dos ganhadores é realizada por meio da computação da solução ótima do problema de maximização a partir do uso do servidor de otimização NEOS [53]. Os resultados publicados corroboram a efetividade do mecanismo em redes orientadas a eventos. Todavia, tal abordagem não é genérica. Logo, provavelmente essa não pode ser facilmente estendida para outros cenários de aplicação das redes de sensores e atores sem fio. Além disso, parece que o protocolo implementado é dependente da disponibilidade de conexão a Internet para a submissão do modelo de otimização formulado em AMPL [68] e formatado em XML ao serviço NEOS, de tal forma que o mecanismo sugerido não é totalmente localizado, e ainda é suscetível a maiores tempos de resposta devido a dependência que se introduz pelo emprego do referido servidor de otimização.

### 3.5 Baseados em Abordagens Biológicas

A divisão de trabalho é fundamental para a organização das sociedades de insetos, e atribui-se a essa propriedade um dos principais fatores de seu sucesso ecológico [179]. Em uma colônia de insetos, um trabalhador não executa todas as tarefas, mas ao contrário, esse se especializa em um conjunto de atividades de acordo com a sua morfologia, idade ou oportunidade. Sugere-se que esse padrão de divisão de trabalho no qual atividades diferentes são executadas simultaneamente por grupos de indivíduos especializados é mais eficiente do que se essas fossem executadas sequencialmente por indivíduos não especializados [24]. Ao mesmo tempo, uma característica marcante da divisão de trabalho em sociedades de insetos é a sua plasticidade, isto é, o grau de flexibilidade comportamental possuído por cada indivíduo da colônia. É essa

propriedade que permite a uma colônia responder a mudanças nas condições internas e externas por meio do ajuste da taxa de trabalhadores individuais engajados nas diversas tarefas, e portanto, se adaptar a novas condições de forma dinâmica e autônoma [136].

Tanto um inseto quanto uma colônia de insetos são sistemas complexos [35]. Todavia, a complexidade individual de um inseto não é suficiente para explicar a complexidade que uma colônia pode apresentar. Isso impede o estabelecimento de uma relação causal entre o comportamento individual e o desempenho coletivo da colônia, o que impossibilita a elucidação do mecanismo subjacente que permite a cooperação em tais sociedades. Alguns dos mecanismos que embasam tal cooperação são determinados geneticamente. Porém, muitos aspectos das atividades coletivas desempenhadas pelos insetos derivam da sua auto-organização.

Teorias de auto-organização ressaltam que instâncias de um comportamento coletivo complexo podem emergir a partir de interações entre indivíduos que exibem um comportamento simples. Mais especificamente, a propriedade de auto-organização consiste de um conjunto de mecanismos dinâmicos pelos quais estruturas aparecem no nível global do sistema a partir de interações entre seus componentes de nível inferior [24]. As regras que especificam as interações entre tais unidades são executadas somente com base em informação localizada, o que permite ao sistema a manifestação de um comportamento emergente e adaptativo.

A descoberta de que o conceito de auto-organização pode explicar o funcionamento cooperativo das sociedades de insetos possibilita a transferência de conhecimento a respeito de tais sociedades para problemas de engenharia e da ciência da computação [25]. Uma das mais importantes características das sociedades de insetos é que elas resolvem problemas de uma maneira flexível e robusta. A flexibilidade permite a adaptação a mudanças ambientais, enquanto que a robustez dota a colônia com a habilidade de continuar funcionando mesmo que elementos individuais venham a falhar no desempenho de suas tarefas. Outra característica consiste no fato de que tais sociedades possuem capacidades cognitivas limitadas; então, abre-se a possibilidade de se projetar agentes artificiais com repertório comportamental simplificado que possam imitar o seu comportamento.

A modelagem de sociedades de insetos por meio do conceito de auto-organização pode auxiliar no projeto de sistemas artificiais que se auto-organizam para resolver problemas, os são quais denominados *swarm intelligence systems* [21]. Tal abordagem consiste no projeto de sistemas artificiais robustos, flexíveis, descentralizados e adaptativos, que se inspiram na metáfora do comportamento coletivo das sociedades de insetos para a solução de problemas. No entanto, tais sistemas são difíceis de projetar, pois que os mecanismos empregados na resolução de problemas não são pré-definidos. Ao contrário, esses são emergentes, e resultam de interações entre os indivíduos, e entre esses e o ambiente. Por isso, o uso desse paradigma requer um profundo conhecimento de quais comportamentos individuais devem ser implementados bem como de quais interações são necessárias para produzir o comportamento global desejado [24].

Tal metáfora possibilita o surgimento de uma nova forma de se projetar sistemas inteligentes, nos quais o comportamento autônomo e emergente e o funcionamento distribuído substituem



as abordagens baseadas em controle, pré-programação e centralização [24]. A aplicação dessa metáfora em sistemas fabricados pelo homem tem se tornado um importante assunto de pesquisa nos últimos anos. Especialmente nas áreas de sistemas autônomos e redes de comunicação [99].

A inteligência empregada pela natureza na resolução de problemas de auto-organização e o estudo de sistemas biológicos oferecem alternativas não convencionais para problemas de pesquisa na área da computação. Logo, a adaptação e o mapeamento de tais soluções para o contexto tecnológico podem fornecer soluções potenciais para redes WSN autônomas e auto-organizáveis [99]. Espera-se que o uso de tais métodos permitam a emergência de um comportamento totalmente auto-organizado da rede por meio do emprego da informação disponível localmente em cada nó.

No contexto da organização do trabalho em sociedades de insetos, destaca-se que a alocação de tarefas é o processo que ajusta o número de insetos engajados em uma dada tarefa de uma forma apropriada a situação corrente da colônia [73]. Uma vez que esse processo é equivalente ao problema da alocação dinâmica de tarefas no âmbito da coordenação ator-ator, nessa seção são apresentados mecanismos localizados que se baseiam no reuso do paradigma *swarm intelligence*.

### 3.5.1 O Modelo de Limiar de Resposta

Em [26] [27] [168] foi apresentado um modelo que conecta o nível de plasticidade individual de um inseto à capacidade de uma colônia responder a eventos ambientais. Nesse modelo, o qual depende de *response thresholds* (ou limiares de resposta), cada indivíduo da colônia possui um limiar de resposta para uma dada tarefa. Nesse contexto, um indivíduo da colônia se engaja no desempenho de uma tarefa quando a intensidade do estímulo associado a essa supera o seu limiar de resposta individual, o qual é hormonalmente regulado. Adicionalmente, a manifestação dessa resposta comportamental é modulada pelas condições ambientais que determinam a probabilidade de um inseto encontrar a tarefa a ser executada [136]. Na Figura 13 (adaptada de [136]), pode-se visualizar o modelo de mecanismo regulador que determina a alocação de tarefas em colônias de abelhas (contudo, evidências experimentais indicam que esse modelo pode ser generalizado para outras espécies de insetos [24]). Cabe enfatizar que tal modelo é baseado no conceito de *feedback*, o qual parece ser onipresente no contexto científico em geral.

Tal modelo simplificado de limiar de resposta explica como a flexibilidade comportamental da colônia resulta da flexibilidade comportamental individual [26] [25] [27] [169]. Evidências experimentais que corroboraram esse modelo foram identificadas em estudos conduzidos com formigas e abelhas [135] [136] [57] [58] [146] [147]. Segundo esses resultados, constata-se que insetos que possuem um limiar de resposta menor respondem à níveis menores de estímulo do que insetos que possuem um limiar de resposta maior. Verifica-se ainda que o desempenho

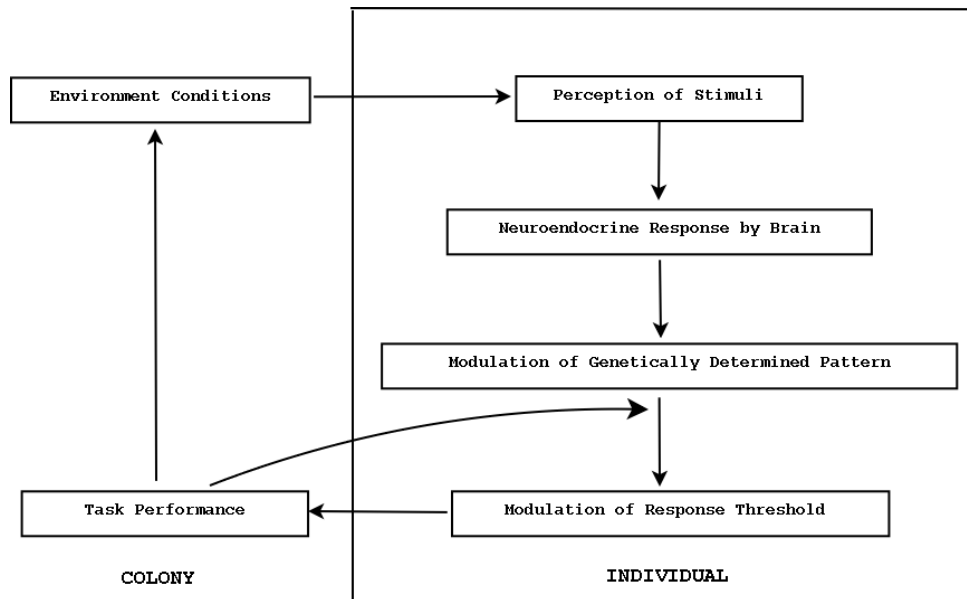


Figura 13 – Modelo de Mecanismo Regulador do Comportamento Laboral Baseado em Resultados Obtidos com Abelhas

da tarefa reduz o limiar de resposta, enquanto que o não desempenho dessa acarreta em um aumento do referido limiar [168]. Uma característica importante desse modelo biológico para o contexto do projeto de sistemas computacionais é que modelos baseados no limiar de resposta mostram que a coordenação para o desempenho de tarefas não necessita estar codificada no comportamento de cada indivíduo [24], de maneira tal que esse possibilita a alocação de tarefas de forma dinâmica.

Constata-se na literatura a existência de duas formulações para o modelo de limiar de resposta: o fixo e o variável. O modelo de limiar de resposta fixo [27] assume que os limiares de resposta dos indivíduos são fixos no tempo. No entanto, resultados obtidos em estudos conduzidos em colônias de abelhas sugerem que esse modelo não é consistente [34] [137], e além do mais, são válidos somente para intervalos de tempo relativamente curtos. Tendo como objetivo superar tal limitação, foi proposto o modelo de limiar de resposta variável [168]. Esse estendeu o modelo anterior, permitindo que tais limiares sejam variáveis no tempo, o que possibilita o surgimento de um processo de reforço que viabiliza uma forma simples de aprendizado: um limiar de resposta diminui quando a tarefa correspondente é desempenhada e aumenta quando a referida tarefa não é desempenhada.

Uma das possíveis famílias de funções de resposta  $T_{\theta}(s)$  para a formalização do modelo de limiar de resposta fixo é expressa pela Equação 3.3 [24]. Essa indica a probabilidade de se desempenhar uma dada tarefa como função da variável intensidade de estímulo  $s$  associado a essa tarefa. Um limiar de resposta  $\theta$  é expresso em unidades da intensidade de estímulo, e é uma variável interna que determina a tendência de um indivíduo a responder a um estímulo  $s$  e a desempenhar a tarefa associada. O significado de  $\theta$  na equação é claro: quando  $s \ll \theta$ , a probabilidade de se engajar no desempenho da tarefa é próximo a 0; quando  $s \gg \theta$ , essa

probabilidade é próxima de 1. Portanto, indivíduos que possuem um pequeno  $\theta$  possuem uma maior chance de responder a um nível menor de estímulo.

$$T_{\theta}(s) = \frac{s^n}{s^n + \theta^n} \quad (3.3)$$

Por sua vez, o modelo de limiar de resposta variável também pode ser exemplificado pela Equação 3.3. Entretanto, nesse modelo o limiar de resposta  $\theta$  é atualizado [170] a cada vez que a tarefa é desempenhada, o que implica no reforçamento do padrão comportamental e portanto resulta na especialização da execução da tarefa. Assume-se que  $\varepsilon$  e  $\varphi$  sejam os coeficientes que descrevem o aprendizado e o esquecimento, respectivamente. No referido modelo, se uma tarefa é executada dentro de um intervalo de tempo  $\Delta t$ , o limiar de resposta é atualizado de acordo com a Equação 3.4. Em caso contrário, esse é atualizado conforme a Equação 3.5.

$$\theta \rightarrow \theta - \varepsilon \Delta t \quad (3.4)$$

$$\theta \rightarrow \theta + \varphi \Delta t \quad (3.5)$$

Contudo, a elaboração de um sistema computacional baseado nesse modelo não é uma tarefa trivial. Isso se deve ao fato de que o sistema artificial resultante não contém um conjunto de comportamentos pré-programados, de forma que não se pode garantir a manifestação de comportamento determinístico no sistema projetado. Ao contrário, o modelo de limiar de resposta pode garantir somente a manifestação de um comportamento estocástico, dado que a função que define o processo decisório computa como saída um valor que indica a probabilidade de um agente se engajar na execução de uma tarefa. Isso visa à simulação das complexas interações ambientais, hormonais e comportamentais que definem o mecanismo regulador da atividade laboral no contexto das sociedades de insetos. Dado tal cenário, podem ser formulados modelos de alocação de tarefas que se baseiam simultaneamente nos mecanismos de mercado e nos mecanismos de regulação laboral dos insetos com o propósito de minimizar o grau de manifestação desse comportamento não determinístico.

Em [106] foi proposto um *framework* para a alocação de tarefas em sistemas constituídos por times de robôs móveis. Esse é constituído por um conjunto de passos genéricos que permitem a um robô resolver de forma autônoma o problema de se engajar ou não no desempenho de uma tarefa. Nesse trabalho, o problema da alocação de tarefas é decomposto em três passos, sendo que essa decomposição é usada na construção de uma formulação genérica baseada em mecanismos de mercado para o problema da coordenação no contexto de múltiplos robôs. Esse conjunto de passos é descrito a seguir:

- Cada robô submete um lance para desempenhar uma tarefa; esse se baseia na percepção da sua adequação para o seu desempenho.
- Um mecanismo baseado em leilão decide qual robô obtém a permissão de desempenhar

essa tarefa.

- O controlador do robô que venceu o leilão desempenha uma ou mais ações para executar essa tarefa.

Tal *framework* foi estendido em [62]. Esse trabalho apresenta um modelo híbrido de alocação de tarefas para o contexto de sistemas multi-agentes, o qual emprega tanto princípios econômicos quanto princípios biológicos na sua operacionalização. Esse é constituído pelo seguinte conjunto de etapas:

- *Should We Bid* - na definição original do modelo de limiar de resposta, a resposta de um inseto ao desempenho de uma tarefa é definida por uma regra de decisão probabilística, a qual é avaliada localmente e de forma autônoma por cada indivíduo da colônia. Entretanto, para que tal modelo possa ser estendido a domínios mais realísticos, faz-se necessário a inclusão de informação específica a um dado domínio no processo de decisão [62]. Essa integração pode ser viabilizada, por exemplo, pelo acréscimo de informações que um dado agente venha a possuir a respeito de suas capacidades, da atividade e do ambiente em particular.
- *Bidding* - a partir do sucesso de abordagens baseadas em mecanismos de mercado para a solução do processo de alocação de tarefas [177] [72], verifica-se uma tendência na literatura de se realizar a aproximação desse paradigma ao modelo de limiar de resposta. Nessa etapa em específico, uma oferta (ou *bid*) consiste na informação de entrada a ser empregada na etapa posterior. Dessa maneira, a formulação matemática que define tal oferta deve fornecer algumas medidas que possam permitir a comparação das ofertas submetidas pelos distintos agentes a uma dada tarefa que deve ser executada.
- *Dominance* - quando múltiplos agentes competem por uma mesma tarefa, é necessário que exista um mecanismo que possa determinar o vencedor. A abordagem empregada na referida extensão é derivada de um mecanismo probabilístico formulado em [47], o qual é fiel ao modelo de limiar de resposta. Logo, tal mecanismo visa à reprodução do comportamento emergente e oportunístico manifestado no interior das colônias de insetos. A Equação 3.6 é uma exemplificação desse mecanismo. Nessa equação, quanto maior for a oferta submetida por um agente  $i$  quando comparada com a oferta submetida por um agente  $j$ , maior é a probabilidade de que o agente  $i$  seja escolhido para desempenhar a tarefa em questão.

$$P(\text{winner} = i) = \frac{Bid_i^2}{Bid_i^2 + Bid_j^2} \quad (3.6)$$

- *Learning* - Conforme [24], o desempenho de uma tarefa por parte de um inseto diminui o limiar de resposta associada a essa, do que resulta a especialização na execução dessa tarefa. Sugere-se que esse procedimento de aprendizado poderia incrementar os limiares

de resposta também para tarefas dissimilares [62], o que poderia reforçar a divisão de trabalho. Para que esse fenômeno possa se manifestar, propõe-se que os valores de limiar de resposta sejam definidos dinamicamente por meio do uso de informação ambiental, de modo que os agentes poderiam responder de forma adaptativa a mudanças nas condições ambientais.

### 3.5.2 Baseado no Modelo de Limiar de Resposta

O mecanismo discutido nessa seção se baseia na avaliação da Equação 3.7. Essa calcula a probabilidade de um ator  $i$  se engajar no desempenho de uma tarefa  $l$ , dado um conjunto de variáveis internas ao ator  $i$  que tentam reproduzir o comportamento laboral de um inseto. Esse comportamento é sujeito tanto a regulação hormonal quanto a modulação ambiental, o que é encapsulado pelo modelo de limiar de resposta. A referida equação é uma adaptação da Equação 3.3 para o contexto de uma WSN. Cabe enfatizar que nos mecanismos apresentados nesse artigo, assume-se que  $n = 2$ . Considera-se que esse é um parâmetro do sistema [47], o qual foi usado no modelo de limiar de resposta original [25], e argumenta-se que resultados similares podem ser obtidos com valores de  $n > 1$  [24].

$$P_{il} = \left( \frac{dae_m^n}{dae_m^n + \omega\theta_{il}^n + \frac{1}{\lambda r_{ik}^n + \rho t_{idle_i}^n} + (\alpha d_{im}^n + \beta e_i^n + \gamma t_{end_{ikl}}^n)} \right) \times I_{ikml} \quad (3.7)$$

O entendimento do modelo de limiar de resposta específico definido pela Equação 3.7 é discutido a seguir. No numerador, tem-se o diâmetro da área de eventos ( $dae_m$ ), que na formulação presente, consiste na intensidade de estímulo associada ao desempenho da tarefa  $l$  em uma área de eventos  $m$ . No denominador, encontram-se quatro termos. O primeiro consiste no  $dae_m$ , enquanto que o segundo consiste no limiar de resposta  $\theta_{il}^n$  associado ao estímulo  $dae_m$ . Quanto menor for o valor de  $\theta_{il}^n$ , maior o valor calculado para a probabilidade  $P_{il}$ . O terceiro e quarto termos consistem em expressões que definem o grau de aptidão do ator  $i$  para o desempenho da tarefa  $l$  em  $m$ . No terceiro termo, quanto maior for o raio de atuação do atuador  $k$  conectado ao ator  $i$  e quanto maior for o tempo no qual o ator  $i$  não esteve envolvido na execução de tarefas, menor será o valor calculado para esse termo. Portanto, menor será a contribuição parcial desse valor para o cálculo da probabilidade do ator  $i$  se engajar no desempenho de  $l$ . No que se refere ao quarto termo da referida equação, quanto menor for a distância entre o nó ator  $i$  e a área de eventos  $m$ , quanto menor for a quantidade de energia dispendida pelo nó  $i$  até o momento da computação desse termo e quanto menor for o tempo necessário para que o ator  $i$  via atuador  $k$  possa completar a execução da tarefa  $l$ , menor será o valor calculado para esse termo. Logo, menor será a contribuição do valor computado para esse termo na determinação da probabilidade do ator  $i$  se engajar no desempenho de  $l$ . E finalmente,  $I_{ikml}$  é uma variável binária que indica

a capacidade do ator  $i$  desempenhar uma tarefa  $l$  por meio do atuador  $k$  na área de eventos  $m$ . Assim como nos protocolos baseados no CNP, o valor assumido pela variável  $I_{ikml}$  depende da avaliação da proposição  $\exists i, j, k, m : ((sc_j \subset ae_m) \wedge (d_{aki} \cap sc_j))$ .

A adaptação do modelo de limiar de resposta ao problema da coordenação ator-ator é discutida a seguir. Após a recepção da informação sensoriada por um nó sensor  $n_{s_o}$ , o nó ator  $n_{a_i}$  assume o papel de coletor, enquanto que os nós atores vizinhos a esse assumem o papel de nó comum na rodada atual do protocolo. O coletor retransmite essa informação aos nós comuns, avalia a sua probabilidade de desempenhar a tarefa corrente por meio da computação da Equação 3.7 e compara o valor computado com um número real randômico gerado no intervalo  $[0, 1]$ . Se o valor computado for maior ou igual ao número randômico gerado, o coletor executa a ação no ambiente controlado. Comportamento similar é reproduzido pelos nós comuns, salvo que esses não retransmitem a informação sensoriada para os seus vizinhos.

Entretanto, o modelo de comportamento oportunístico expresso pela Equação 3.7 não consegue garantir deterministicamente o critério de cobertura da área de eventos  $m$ . Assim, com o objetivo de se aumentar o nível de precisão do referido protocolo, foi elaborado o mecanismo descrito a seguir.

### 3.5.3 Baseado em Extensões ao Modelo de Limiar de Resposta

O problema da coordenação ator-ator consiste na definição de um mecanismo que permita a determinação de qual ator ou conjunto de atores deve desempenhar uma dada tarefa. Essa instância do problema da alocação de tarefas não possui uma solução trivial, visto que em uma WSAAN tarefas podem ser criadas dinamicamente a partir das informações sensoriadas do ambiente.

Na subseção anterior, foi proposto um mecanismo para a alocação de tarefas que se baseia no limiar de resposta de um indivíduo a uma dada tarefa. Nesse modelo, um ator decide em qual tarefa se engajar por meio de uma regra de decisão probabilística que leva em conta uma avaliação da tarefa e o seu custo. Frequentemente, essa regra de decisão é empregada em conjunto com uma regra de aprendizado simples, a qual adapta o limiar de resposta de um dado indivíduo a um contexto específico. Entretanto, a definição dessa regra de decisão não é um processo simples. Isso se deve ao fato de que essa deve ser ajustada manualmente com a finalidade de se adequar a um domínio particular. Além do mais, tal regra de decisão pode implicar em um modelo de alocação de tarefas com significativo grau de imprecisão. Sugere-se que um modelo híbrido em seu funcionamento pode minimizar essa imprecisão [74].

A adaptação do modelo híbrido proposto em [62] ao problema da coordenação ator-ator é discutida a seguir. Analogamente ao que ocorre no protocolo proposto com base no Modelo de Limiar de Resposta puro, o nó ator  $n_{a_i}$  que recebe o pacote com a informação sensoriada pelo nó sensor  $n_{s_o}$  assume o papel de coletor na rodada atual do protocolo, enquanto que os nós

atores vizinhos a esse assumem o papel de nó comum.

Inicialmente, o nó coletor calcula o número de nós comuns necessários a execução da ação de controle sobre o ambiente na rodada corrente do protocolo. Isso é realizado através da comparação do raio de propagação da área de eventos com o seu raio de atuação. Em seguida, o coletor retransmite a informação sensoriada aos nós comuns, e transita para o estado associado a etapa *Dominance*. Nessa etapa, o coletor avalia a sua probabilidade de desempenhar a tarefa atual através da computação da Equação 3.7, e após realiza a comparação do valor computado com um número real randômico gerado no intervalo  $[0, 1]$ . Se o valor computado for maior ou igual ao número randômico gerado, o coletor executa a ação no ambiente controlado. Ainda nessa etapa, o coletor seleciona o subconjunto mínimo de atores que garantem o critério de cobertura da área de eventos  $m$ . Para tanto, ele ordena os lances que foram submetidos a partir da computação das Equações 3.8 e 3.2 em ordem descendente. A interpretação dos termos da função de custo definida pela Equação 3.8 é análoga a interpretação da função de custo definida pela equação 3.1, com exceção do acréscimo da variável  $t_{idle_i}$  ao terceiro termo. Com essa modificação, pode-se verificar que quanto maior for o raio de atuação do atuador  $k$  conectado ao ator  $i$  e quanto maior for o tempo no qual o ator  $i$  não esteve envolvido na execução de tarefas, menor será o valor calculado para esse termo, e por conseguinte, maior será o lance submetido.

$$f c_i = \frac{\alpha d_{im} + \beta e_i + \gamma t_{end_{ikl}}}{\alpha + \beta + \gamma} + \frac{1}{\lambda r_{ik} + \rho t_{idle_i}} \quad (3.8)$$

Contudo, o coletor não notifica diretamente os nós comuns de que esses foram selecionados para o desempenho da tarefa. Nesse caso, o mecanismo de determinação dos nós mais aptos executa um passo adicional, o qual consiste na computação da Equação 3.6 para cada par de lances  $i$  e  $j$ . Essa conduz o processo de seleção dos nós atores mais capazes de forma probabilística, com o objetivo de reproduzir o comportamento não determinístico do modelo de limiar de resposta. Após a computação da referida probabilidade, o valor computado é comparado com um número real randômico gerado no intervalo  $[0, 1]$ . Se o valor computado for maior ou igual ao número randômico gerado, o coletor envia uma mensagem ao nó corrente que foi selecionado probabilisticamente, o que lhe habilita a executar a ação no ambiente controlado.

Quando um nó comum recebe a informação retransmitida pelo coletor, a qual não é retransmitida aos seus vizinhos, ele entra no estado associado a etapa *Should We Bid*. Nessa etapa, o nó avalia a sua probabilidade de desempenhar a tarefa corrente por meio da computação da Equação 3.7 e compara o valor computado com um número real randômico gerado no intervalo  $[0, 1]$ . Se o valor computado for maior ou igual ao número randômico gerado, o nó passa para o estado *Bidding*. Nesse estado, o nó submete um lance (o qual contém a totalidade das variáveis que descrevem o seu grau de aptidão) e envia esse para o nó coletor. O referido lance é empregado no processamento da etapa *Dominance*, a qual foi explicitada no parágrafo anterior.

### 3.5.4 Trabalhos Relacionados

Conforme mencionado anteriormente, parece que ainda existe espaço para contribuições no que se refere à questão da coordenação ator-ator em redes de sensores e atores. A seguir, discutem-se algumas propostas baseadas em princípios derivados da teoria biológica que endereçam esse problema.

Em [99] é formulado um modelo de divisão de trabalho baseado em decisões probabilísticas. Nesse modelo, os agentes (sensores e os atores) se adaptam de forma probabilística para a execução de uma tarefa dentro de um conjunto de tarefas. A arquitetura dessa solução explora interações entre os nós que estão dentro de um intervalo geográfico limitado, de modo que as interações locais são suficientes para induzir a divisão de trabalho no nível global da rede [61]. Dessa forma, não é necessário que se conheça o ambiente particular ou que atividades estão sendo exercidas pelos outros nós da rede. Todavia, a expressão que define a probabilidade de um agente executar uma dada tarefa não leva em conta o seu grau de aptidão, nem as condições ambientais às quais os agentes estão sujeitos. Ademais, os agentes conhecem *a priori* a lista de possíveis atividades que eles podem desempenhar, o que pode sugerir uma limitação da aplicabilidade de tal mecanismo a outros cenários.

Em outro trabalho [183] são propostos modelos de coordenação para as três esferas: sensor-sensor, sensor-ator e ator-ator. Peculiarmente, são sugeridas duas soluções para o problema da coordenação caso ator-ator. Uma dessas é denominada *action-first*. Nesse mecanismo, quando um ator recebe informações sensoriadas do ambiente, ele executa a ação imediatamente sem recorrer ao estabelecimento de um processo de negociação com outros atores, o que possibilita a redução da latência entre o momento em que ocorre a captura do evento ambiental e o momento no qual a ação é efetivamente desempenhada. Nesse esquema, o qual se baseia num mecanismo semanticamente equivalente ao modelo de limiar de resposta, cada ator decide independentemente se atua ou não no ambiente. Esse trabalho apresenta um estudo de caso que explora esse mecanismo de coordenação. Contudo, não foram publicados resultados que possam corroborar a sua efetividade.



## 4 WSANSim: Um Ambiente de Simulação para WSN

### 4.1 Definição do Problema

Diversos protocolos foram propostos para as redes de sensores sem fio [8]. Entretanto, visto que os requisitos das redes de sensores e atores sem fio introduzem novos desafios de projeto em complemento aos requisitos daquelas, provavelmente tais protocolos não são adequados as suas características únicas [7]. Logo, é razoável afirmar que os protocolos originariamente projetados para as redes de sensores terão que provavelmente ser adaptados ou completamente redesenhados para o contexto das redes de sensores e atores sem fio.

As técnicas comumente utilizadas para a análise de desempenho de sistemas são experimentos em laboratório, métodos analíticos e simulação computacional [100]. Mesmo sendo um método imperfeito, a técnica de simulação computacional é a abordagem mais usada para a avaliação de desempenho de novas propostas de pesquisa na área de redes de sensores e atores sem fio. Além disso, projetistas frequentemente necessitam obter alguma informação a respeito da viabilidade de um projeto antes de investir significativo volume de dinheiro e de tempo no seu desenvolvimento. Tal afirmação é especialmente verdadeira no contexto de tais redes, nas quais os nós sensores devem ser adquiridos em larga quantidade a um custo elevado e nas quais os nós atores podem ser adquiridos a um alto custo individual. E mesmo que tais dispositivos estejam disponíveis, a testagem da rede no ambiente desejado pode ser uma tarefa difícil e demorada.

Como consequência, um ambiente de simulação mais realístico para a avaliação de protocolos focados nas redes de sensores e atores sem fio poderia trazer benefícios em termos de tempo e custo. Contudo, de acordo com a análise realizada, não existe um ambiente de simulação para WSN bem estabelecido na comunidade de pesquisa. Por tal motivo, o WSANSim foi concebido e implementado.

### 4.2 Trabalhos Relacionados

Tendo em vista o desenvolvimento de novos protocolos para as redes de sensores bem como a sua avaliação de desempenho, diversos ambientes de simulação foram estendidos. Outros ambientes de simulação foram projetados a partir do zero. Tais ambientes permitem a simulação do comportamento das redes de sensores. Contudo, parecem não incorporar determinadas abstrações de software derivadas da Teoria de Controle que permitam reproduzir o comportamento

de uma rede de sensores e atores com um certo grau de precisão.

#### 4.2.1 ns-2

O ns-2 [117] é uma ferramenta de simulação popularmente empregada na implementação de protocolos de rede e de algoritmos distribuídos. O ns-2 é um simulador orientado a objetos, cuja abordagem modular permite a sua extensibilidade. As simulações são baseadas em uma combinação de código C++ e OTcl [171]. Usualmente, a linguagem C++ é empregada para a implementação de protocolos e para a realização de extensões das bibliotecas do ns-2, enquanto que a linguagem OTcl é usada para a criação e para o controle do ambiente de simulação.

Mais especificamente, o ns-2 é um simulador de eventos discretos, no qual o avanço do tempo depende da temporização dos eventos que são mantidos pelo escalonador [10]. Um evento é um objeto na hierarquia de classes C++ que possui um identificador único, um tempo de escalonamento e um ponteiro para um objeto que manipula o evento. O escalonador mantém uma lista ordenada que armazena os eventos que devem ser disparados e os aciona um por um, invocando o manipulador associado ao evento corrente. Em sua versão corrente, o ns-2 é *single-threaded*, de maneira que somente um evento pode ser disparado em uma dada instância de tempo virtual.

O referido simulador possui uma biblioteca de protocolos de rede [128]. Existem duas hierarquias de classe: a hierarquia C++ compilada e a hierarquia OTcl interpretada, com uma correspondência de um para um entre ambas as hierarquias. A hierarquia C++ compilada permite eficiência na simulação, a qual é empregada na operação dos protocolos de rede. Por sua vez, a hierarquia OTcl interpretada é usada nos *scripts* escritos pelos usuários, os quais definem uma topologia de rede particular bem como os protocolos e aplicações que devem ser simulados. O interpretador OTcl, o qual é responsável pela execução dos *scripts*, é o *frontend* que faz uso dos objetos compilados na hierarquia C++ por meio de uma ligação em OTcl que realiza a associação de um objeto OTcl para cada objeto C++.

Todavia, o ns-2 apresenta pontos negativos [80]. O mais conhecido consiste no significativo consumo de recursos computacionais. Como decorrência desse fato, o ns-2 não possui a escalabilidade necessária para a simulação de redes compostas por um número relativamente grande de nós. Logo, ele se mostra inadequado para a simulação de redes de sensores sem fio.

#### 4.2.2 pdns

O pdns (*Parallel/Distributed Network Simulator*) [134] possui como objetivo sobrepujar as limitações do ns-2 no que refere ao requisito escalabilidade. Essa extensão visa ao aumento do desempenho do ns-2 pela distribuição da carga do experimento de simulação em diversas

estações de trabalho conectadas em rede. Por consequência, o pdns pode obter uma paralelização eficiente do processo de simulação ao realizar a distribuição das instâncias de simulação do ns-2 em sub-redes distintas. Entretanto, trabalhos publicados recentemente sugerem que o uso de tal ferramenta pela comunidade de redes de sensores ainda não foi disseminado na mesma proporção que o uso da versão convencional do ns-2.

### 4.2.3 SensorSim

O SensorSim [122] [123] emprega o código do ns-2 como base e o amplia em três maneiras. Em primeiro lugar, ele inclui um novo modelo de energia para redes de sensores, o qual leva em conta cada um dos componentes de *hardware* que poderia necessitar de energia da bateria para operar. Em segundo lugar, ele introduz o conceito de canal de sensoriamento, o qual pode ser visto como uma analogia ao canal de rede sem fio. Esse é um *container* no qual uma interface de rede sem fio pode sinalizar outras interfaces ao iniciar um evento (transmissão de um pacote) ou detectar um evento (recepção de um pacote). O canal de rede sem fio usa o ar como meio de transporte dos sinais de rádio. O conceito de canal de sensoriamento é similar ao conceito de canal de rede sem fio, com exceção do fato de que os dispositivos de sensoriamento são passivos. Eles somente detectam sinais gerados por entidades externas que são propagados pelo canal de sensoriamento. Em terceiro lugar, ele possibilita a emulação de uma rede de sensores. Ou seja, ele implementa um mecanismo que possibilita a interação com aplicações externas, o que permite ao SensorSim interagir com nós sensores reais. Entretanto, no momento da escrita desse trabalho, o código do SensorSim não está mais disponível publicamente [115].

### 4.2.4 J-Sim

O J-Sim [159] [158] é simulador de uso geral baseado na linguagem Java que foi modelado após o ns-2. Diferentemente desse, aquele é implementado na camada superior de uma arquitetura de software baseada em componentes [166] denominada *Autonomous Component Architecture (ACA)*. Os elementos básicos dessa arquitetura são os componentes, os quais se comunicam entre si pelo envio e pela recepção de dados em suas portas. No nível superior da ACA, existe um *framework* de interconexão comutado a pacotes genérico denominado INET. Ambos foram implementados em Java, sendo que o código resultante é denominado J-Sim. Similarmente ao ns-2, o J-Sim é um ambiente de simulação com duas linguagens, no qual as classes escritas em Java são conectadas entre si por meio da linguagem Tcl.

O *framework* de simulação para redes de sensores construído no J-Sim é derivado do modelo proposto pelo SensorSim. Os nós sensores detectam sinais gerados pelos nós *target* no canal de sensoriamento e enviam as informações detectadas para o nó escoadouro por meio do emprego

do canal de rede sem fio. Portanto, existem dois modelos diferentes para a propagação de sinais no referido *framework*: um modelo de propagação dos sinais sensorizados e um modelo de propagação dos sinais de rádio. Assim como no caso do SensorSim, o J-Sim também permite a emulação de uma rede de sensores.

O simulador J-Sim introduz alguns avanços sobre o ns-2. O mais importante deriva do fato de que a arquitetura baseada em componentes é mais escalável em termos de uso de memória do que o modelo orientado a objetos empregado pelo ns-2. Dessa forma, uma maior quantidade de nós pode ser empregada nos experimentos por ventura conduzidos nessa plataforma (conforme resultados publicados em [158]). Provavelmente, tal fato deve ser decorrência do mecanismo de *garbage collection* usado pela linguagem Java para a recuperação de memória não usada [55]. Essa funcionalidade não é encontrada em ambientes C++, nos quais a memória não usada deve ser liberada explicitamente pelo programador. Contudo, mesmo sendo mais escalável do que o ns-2 em termos da métrica uso de memória, o J-Sim ainda sofre das ineficiências derivadas da máquina virtual Java. Dessa forma, o ns-2 apresenta desempenho superior no que se refere à métrica tempo de processamento [158].

#### 4.2.5 Ptolemy

O Ptolemy [56] é um projeto em andamento da *Berkeley University* que estuda a modelagem, a simulação de eventos discretos e o desenvolvimento de sistemas concorrentes, de tempo real e embarcados. O princípio básico subjacente ao Ptolemy consiste na sua habilidade de empregar múltiplos modelos de computação (tempo-contínuo, fluxo de dados e máquina de estados finitos) em um ambiente de projeto hierárquico e heterogêneo. A partir de extensões ao Ptolemy, foi construído o VisualSense [19]. Esse é um *framework* de modelagem e simulação que permite o projeto, a simulação e a visualização de redes de sensores. Assim como no caso do SensorSim e do J-Sim, o VisualSense também implementa os canais de sensoriamento e de rede sem fio. Porém, o VisualSense não possui módulos que possibilitem a simulação de redes de sensores e atores sem fio.

#### 4.2.6 Mannasim

O *framework* Mannasim [151], da mesma forma que o SensorSim, emprega o código base do ns-2 e o estende para simular o comportamento de uma rede de sensores. Todavia, o Mannasim está focado no nível da aplicação. Ele fornece estruturas padronizadas para nós sensores comuns, nós *cluster head* e nós escoadouro (ou ponto de acesso). Contém ainda módulos para a geração de dados, para o processamento de dados e para o fornecimento de energia. Da mesma forma que o SensorSim, o Mannasim enfrenta o problema da escalabilidade.

#### 4.2.7 SENSE

O SENSE (*Sensor Network Simulator and Emulator*) [45] tenta implementar o mesmo nível de funcionalidade disponibilizado pelo ns-2. Entretanto, ao mesmo tempo tenta minimizar a interdependência entre os módulos verificada no ns-2, a qual é derivada do seu projeto orientado a objetos. Da mesma forma que o J-Sim, o SENSE adota uma arquitetura baseada em componentes. O SENSE foi escrito em C++, na camada superior do COST (*Component Oriented Simulation*), um simulador de eventos discreto de uso geral que foi escrito em Comp++, o qual por sua vez é uma extensão orientada a componentes da linguagem C++. Contudo, o referido simulador não consegue implementar todas as funcionalidades existentes no ns-2, principalmente devido ao relativamente considerável número de extensões que foram baseadas nesse.

#### 4.2.8 SENS

O SENS (*Sensor, Environment and Network Simulator*) [165] é um simulador baseado em componentes escrito em C++. Ele consiste de diversos nós sensores simulados interagindo com o componente *environment*. Cada nó sensor consiste de três componentes: *application*, *network* e *physical*. Esses componentes são conectados entre si para simular aplicações específicas para redes de sensores, cujo código pode ser distribuído para nós sensores reais. Todavia, o SENS é menos customizável do que os outros simuladores.

#### 4.2.9 Sidh

Sidh [37] é um simulador baseado em componentes escrito em Java. Ele é composto por módulos que se enquadram em um número de diferentes categorias, dentre as quais incluem-se: *Simulator*, *Event*, *Medium*, *Propagation Models*, *Environment*, *Node*, *Transceiver*, *Protocols* e *Applications*. Seus autores argumentam que o Sidh é eficiente e escalável. Contudo, ainda não foram publicados resultados sobre o seu desempenho que possam corroborar tal afirmação.

#### 4.2.10 BAKARA

BAKARA [98] foi o único simulador identificado na literatura com foco em redes de sensores e atores sem fio. Ele foi construído a partir da combinação do código do simulador de rede OMNet++ [172] e da biblioteca de simulação de corpos rígidos *Open Dynamics Engine* (ODE) [153]. O OMNet++ é um simulador de eventos discretos implementado em C++, o qual simula

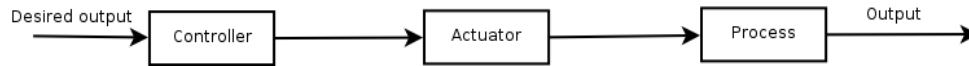


Figura 14 – Diagrama de Blocos de um Sistema de Controle a Malha Aberta

modulos que podem enviar mensagens entre si por meio de canais de comunicação. Cada mensagem representa um evento que é armazenado no escalonador do OMNet++. Analogamente ao que ocorre no ns-2, cada pacote é um evento que é armazenado pelo escalonador. Por sua vez, a ODE permite a simulação do movimento de corpos rígidos. Tais abstrações de software são usualmente empregadas na modelagem da cinética de corpos rígidos na área de robótica [97]. A mencionada biblioteca fornece primitivas que permitem definir a massa, o momento de inércia e a velocidade de um corpo rígido, bem como a aplicação de forças aos objetos modelados. Contudo, até o momento da escrita desse trabalho, a grande maioria dos protocolos que foram propostos para redes de sensores e atores sem fio se utilizaram do ns-2 ou do J-Sim para a avaliação de seus resultados.

### 4.3 Arquitetura

Um sistema de controle é um conjunto de componentes interconectados que formam uma configuração de sistema que produz uma resposta desejada [60]. Tipicamente, tal sistema é constituído pelos seguintes elementos: sensores, controladores e atuadores [22]. Normalmente, um sistema de controle é projetado por meio do uso de uma das seguintes estratégias: malha aberta (ver Figura 14 - adaptado de [60]) ou malha fechada (ver Figura 15 - adaptado de [60]) [118] [60]. Um sistema de controle a malha aberta utiliza um atuador para controlar diretamente um processo físico sem usar retroação. Ao contrário desse, um sistema de controle a malha fechada controla diretamente um processo físico por meio da comparação do valor de saída obtido via retroação com o valor de saída desejado do sistema, de modo que a ação de controle é dependente da saída do processo físico que está sendo controlado. Em maior grau de detalhe, um sistema de controle a malha fechada realiza o sensoriamento da operação de um sistema, compara a grandeza sensoriada com o comportamento desejado do sistema, computa as ações corretivas baseadas no modelo de resposta do sistema a entradas externas e atua no sistema para produzir a modificação esperada no seu comportamento. Essa malha de retroação, constituída de sensoriamento, computação e atuação, é um dos conceitos fundamentais da área de controle automático de sistemas dinâmicos [113].

Sensores e atuadores são os componentes críticos de todo sistema de controle a malha fechada, o qual pode receber a denominação de sistema mecatrônico [22]. A arquitetura típica de um sistema mecatrônico pode ser visualizada na Figura 16 (adaptado de [22]). Tal categoria de sistema consiste na completa integração dos conhecimentos produzidos nas áreas de engenharia e da computação para o projeto e para a manufatura de sistemas de controle [23] [32].

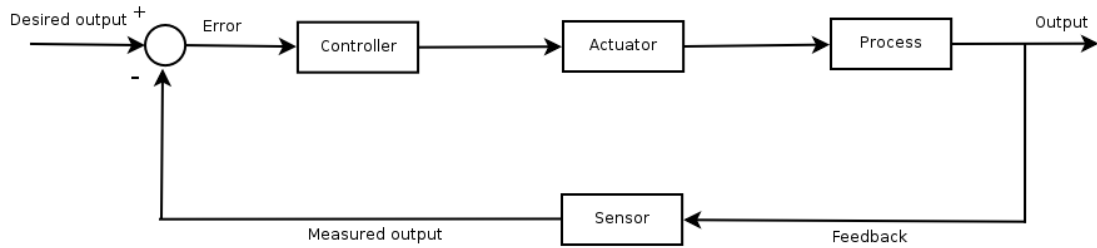


Figura 15 – Diagrama de Blocos de um Sistema de Controle a Malha Fechada

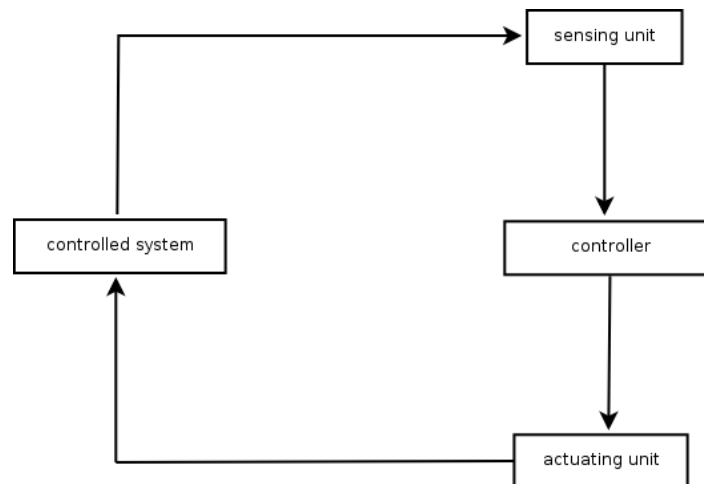


Figura 16 – Arquitetura de um Sistema Mecatrônico

Dado que um sistema mecatrônico consiste na implementação das propriedades abstratas de um modelo matemático de um sistema de controle particular [23], constata-se por analogia que um sistema mecatrônico é composto pelos seguintes componentes: uma unidade de sensoria-mento, uma unidade de controle e uma unidade de atuação [22]. A unidade de sensoriamento pode ser tão simples quanto sendo composta por apenas um sensor, ou pode consistir-se de um conjunto de componentes adicionais. Um sensor é um dispositivo que quando exposto a um fenômeno físico (temperatura, força, pressão, etc.) produz um sinal de saída (elétrico, mecâ-nico, magnético, etc.) proporcional à intensidade desse estímulo. A unidade de controle, a partir da leitura da informação obtida da unidade de sensoriamento, toma decisões baseadas em algum algoritmo de controle e envia comandos para a unidade de atuação. Por sua vez, a unidade de atuação consiste de um atuador e opcionalmente de uma fonte de alimentação e um mecanismo de acoplamento cuja a arquitetura característica pode ser visualizada na Figura 17 (adaptado de [23]). Um atuador recebe um comando de controle (na maioria das vezes, na forma de um sinal elétrico) que foi enviado pelo controlador e produz uma saída no sistema físico por meio da geração de força, movimento, fluxo, etc. Esse dispositivo é usualmente usado em conjunção com o mecanismo de acoplamento, o qual é responsável por atuar como uma interface entre o atuador e o sistema controlado. Mais especificamente, no contexto de uma WSN, um atua-dor é um dispositivo que converte um sinal de controle elétrico em uma ação física, o qual se

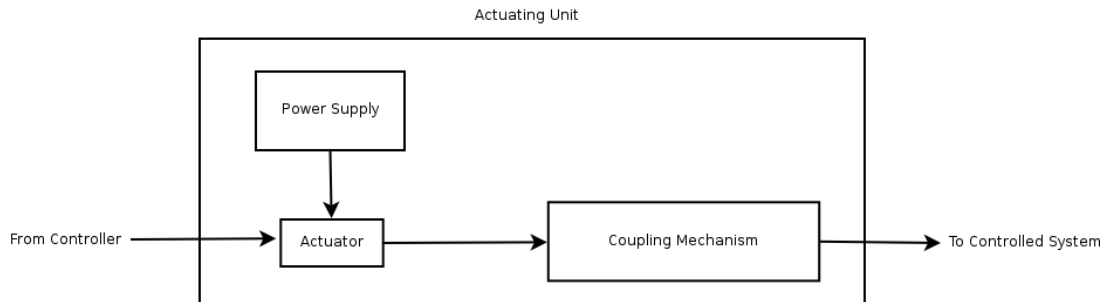


Figura 17 – Arquitetura de uma Unidade de Atuação

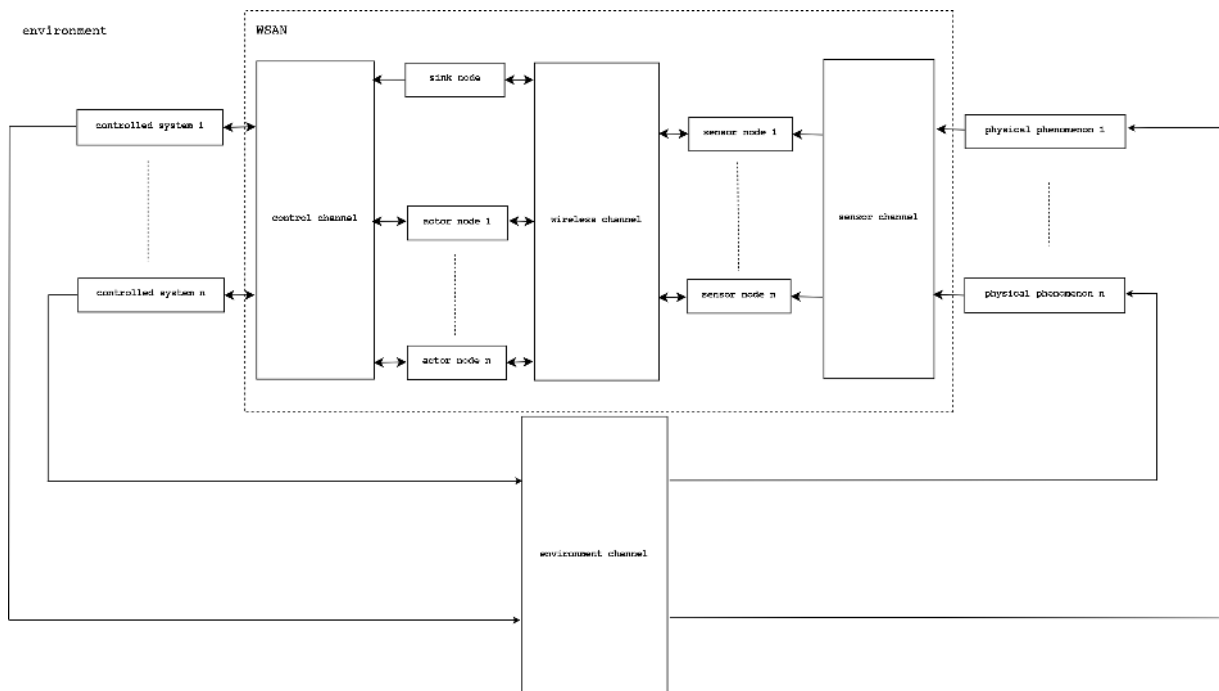


Figura 18 – Arquitetura do *Framework* de Simulação

constitui no mecanismo pelo qual um nó ator atua no ambiente físico [7].

A arquitetura de simulação proposta nesse trabalho emprega uma visão de WSA enquanto sistema de controle (ou sistema mecatrônico), a qual pode ser visualizada na Figura 18. Essa arquitetura, a qual é baseada na estratégia de malha fechada, possui os seguintes componentes genéricos: *physical phenomena*, *sensor channel*, *actor node*, *sensor node*, *sink node*, *wireless channel*, *control channel*, *controlled system* e *environmental channel*. A arquitetura do WSANSim reutiliza a idéia do canal de sensoriamento, a qual foi introduzida pelo SensorSim, e aplica as mesmas idéias ao canal de controle. Ademais, ambos os canais possibilitam que os fenômenos físicos sejam sensoriados e influenciados por meio de ações, o que torna possível a criação de uma arquitetura em malha fechada. Devido a sua generalidade, a arquitetura sugerida para a simulação de uma WSA poderia ser implementada como um extensão de qualquer outro tipo de simulador de rede.



## 4.4 Implementação

O ns-2 é a ferramenta de software mais usada para a simulação de redes sem fio *ad hoc* [80]. Particularmente, as extensões produzidas pelo projeto Monarch [127] da *Rice University*<sup>1</sup> para o ns-2 disponibilizam vários módulos que simulam redes de comunicação sem fio, os quais são necessários para a implementação da arquitetura proposta. Portanto, o ns-2 foi escolhido como código base para o desenvolvimento do *framework* de simulação para WSAAN, embora não seja desconhecido o problema da falta de escalabilidade desse simulador. Todavia, trabalhos publicados recentemente no contexto das redes WSAAN em sua maioria ainda empregam o ns-2 para a avaliação de desempenho de suas propostas [150] [174] [96] [75].

O WSANSim estendeu módulos tanto do ns-2 quanto do projeto Monarch com o objetivo de possibilitar a comunicação via rede sem fio entre suas entidades de simulação. Essas extensões podem ser visualizadas com um maior grau de detalhe no diagrama de classes em UML apresentado na Figura 19. De acordo com esse diagrama, tal comunicação é viabilizada por meio das funcionalidades disponibilizadas pelo canal e pelas interfaces de rede sem fio, pelos modelos de propagação de rádio e pelos protocolos de enlace e de roteamento desenvolvidos pelo projeto Monarch. No caso dos protocolos da camada de enlace, argumenta-se que os protocolos clássicos baseados em contenção não são apropriados para a comunicação sensor-ator e ator-ator em tempo real, visto que tais protocolos de acesso ao meio empregam mecanismos baseados em *handshaking*, o que pode implicar no aumento da latência na transmissão dos pacotes [7]. No entanto, o ns-2 disponibiliza somente dois protocolos de enlace para a comunicação via rede sem fio: o 802.11 [52], o qual se baseia no esquema RTS/CTS/DATA/ACK para envio em modo *unicast* ou no esquema de envio direto em modo *broadcast*, e o TDMA, o qual aloca *slots* de tempo distintos para os nós enviarem e receberem pacotes. O protocolo TDMA foi adicionado ao ns-2 em 2000, e ainda está em estágio experimental [37] [128]. Assim, emprega-se o protocolo 802.11 na camada de enlace do WSANSim, o qual também é empregado pelo *framework* de simulação de redes de sensores acrescentado ao J-Sim. Por sua vez, no caso dos protocolos de roteamento, a entrega de pacotes em tempo real é um requisito fundamental da comunicação sensor-ator [108]. Contudo, não estão disponíveis protocolos de roteamento para a referida comunicação na versão corrente do ns-2. Quanto a comunicação ator-ator, essa é similar ao paradigma de comunicação assumido nas redes MANET, o que se deve a existência de um número reduzido de nós atores que possuem uma maior quantidade de recursos [7]. Como resultado, os protocolos de roteamento desenvolvidos para as redes MANET tais como o DSDV, o DSR, o TORA e o AODV [2] podem ser empregados para que se atenda à restrição temporal inerente a essa modalidade de comunicação. O ns-2 implementa a funcionalidade desses protocolos de roteamento *ad hoc* [128]. O AODV é um protocolo de roteamento reativo e escalável que minimiza o *overhead* que é necessário para a manutenção das tabelas de roteamento por meio

<sup>1</sup>Outrora conhecido como CMU Monarch Project (<http://www.monarch.cs.emu.edu/>).

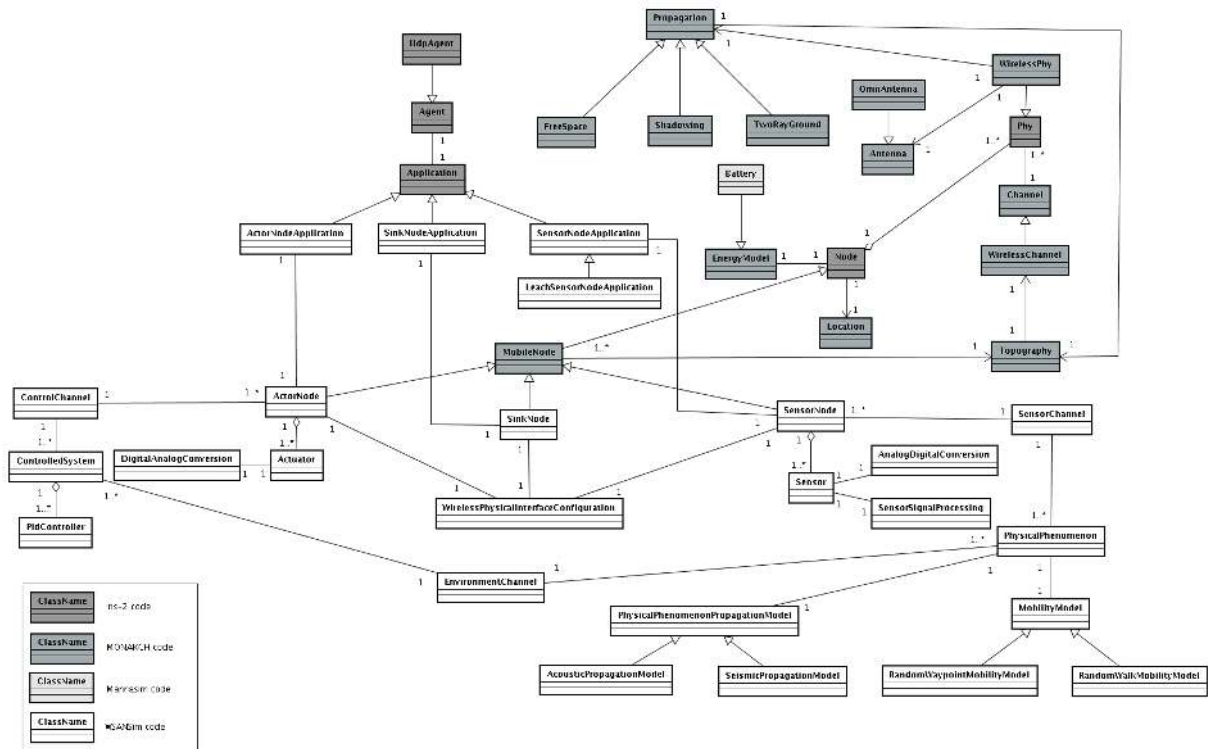


Figura 19 – Diagrama de Classes em UML do WSANSim

do uso de um mecanismo que realiza a determinação das rotas sob demanda [125], o que pode implicar na redução do consumo de energia e no aumento do tempo de vida da rede. Por tal motivo, emprega-se o protocolo AODV na camada de roteamento do WSANSim. Esse também é empregado pelo *framework* de simulação de redes de sensores incluído no J-Sim.

No processo de extensão desses módulos algumas modificações se fizeram necessárias. O código das classes *Mac802-11* e *UdpAgent* teve que sofrer pequenas alterações devido a constatação de que pacotes enviados via *broadcast* para o canal de rede sem fio eram perdidos. O código da classe *WirelessChannel*, a qual implementa a funcionalidade do referido canal, também teve que sofrer modificações devido a falhas de segmentação de memória. Pelos experimentos comparativos que foram conduzidos para a identificação do problema, esse somente se reproduzia em aplicações que se utilizavam de comunicação via *broadcast*, o qual ainda continuava se manifestando após a implementação das alterações acima mencionadas. Em experimentos de simulação com dez instâncias da aplicação FTP nativa do ns-2, a qual se baseia em comunicação em modo *unicast*, esse problema não se manifestou. No entanto, em experimento similar conduzido com aplicações baseadas nas classes de aplicação do WSANSim, as quais empregavam comunicação do tipo *broadcast*, esse problema sempre se manifestava. Ademais, o código das classes *Antenna*, *MobileNode*, *OmniAntenna* e *WirelessPhy* sofreu modificações com o propósito de aumentar o seu grau de modularidade.

As extensões providas pelo WSANSim ao ns-2 fornecem novos elementos na camada de aplicação do ambiente de simulação. Além disso, elementos adicionais extraídos da Teoria de

Controle são incluídos ao ns-2. A modelagem desses elementos foi conduzida com o objetivo primário de se tentar reproduzir detalhes significativos de um sistema de controle distribuído do mundo real, o que permitiria a criação de modelos de simulação com maior grau de fidelidade. Provavelmente, do emprego de tais modelos pode resultar certo incremento na precisão dos resultados obtidos a partir da condução dos experimentos de simulação. Os módulos mais importantes que compõem tais extensões, os quais foram escritos em C++, são descritos a seguir. Eles foram divididos em grupos, de acordo com o propósito principal de cada módulo:

1. Tipos de nós da rede: referem-se às entidades de rede ativas do ambiente de simulação, isto é, as entidades que enviam e recebem pacotes via canal de rede sem fio. Cada nó de rede possui uma posição geográfica, um alcance de comunicação e um modelo de mobilidade (nesse caso específico, é implementado o modelo de mobilidade *Random Walk* [88]).
  - *SensorNode*: instâncias do nó sensor podem ser estáticas ou móveis no terreno de simulação. Eles podem ser conectados ao canal de rede sem fio ou ao canal de sensoriamento. Um nó sensor possui dispositivos de sensoriamento, e possui menor capacidade computacional do que instâncias de nós do tipo escoadouro e do tipo ator.
  - *SinkNode*: instâncias do nó escoadouro podem ser estáticas ou móveis no terreno de simulação. Eles podem ser conectados ao canal de rede sem fio ou ao canal controle.
  - *ActorNode*: instâncias do nó ator podem ser estáticas ou móveis no terreno de simulação. Eles podem ser conectados ao canal de rede sem fio ou ao canal de controle. Um nó ator possui dispositivos de atuação, e possui maior capacidade computacional do que instâncias de nós do tipo sensor.
  
2. Tipos de aplicações: as classes abaixo relacionadas simulam o comportamento das aplicações que podem ser executadas em cada uma das entidades de rede acima relacionadas.
  - *SensorNodeApplication*: instâncias dessa classe estão conectadas em um nó sensor e indicam se esse possui energia suficiente para continuar sensoriando o ambiente e se comunicar via rede sem fio. Essas podem disseminar sinais capturados pelo dispositivo de sensoriamento via canal de sensoriamento para o canal de rede sem fio continuamente, periodicamente ou de forma orientada a eventos.
  - *LeachSensorNodeApplication*: instâncias dessa classe implementam uma versão adaptada do protocolo *Low Energy Adaptive Clustering Hierarchy Protocol* (LEACH) [79]. O LEACH é um protocolo de roteamento hierárquico adaptativo e auto-organizável proposto no âmbito das redes de sensores, cujos requisitos são escalabilidade, comunicação eficiente e eficiência energética. Esse randomicamente elege um pequeno subconjunto de nós sensores, os quais devem desempenhar o papel de

*cluster-head* em cada *cluster* dinamicamente formado, enquanto que os nós membros do *cluster* configuram o seu subsistema de comunicação do estado ativo para o estado dormente a cada fase de cada rodada desse protocolo. O LEACH rotaciona esse papel a um intervalo de tempo  $t_{change}$  fixo, com o objetivo de promover a distribuição do consumo de energia de forma uniforme entre os nós que compõem a rede. O nó *cluster-head* de cada *cluster* realiza a compressão dos dados que foram recebidos de cada membro, e após envia um pacote com os dados agregados para o nó escoadouro com o objetivo de reduzir o número de mensagens que devem ser enviadas pela rede. Na nossa implementação, a operação de agregação dos dados consiste na computação da média aritmética dos valores dos dados obtidos através do mecanismo de sensoriamento de sinais. A partir de resultados de simulação obtidos via experimentos no MATLAB [38], os autores do LEACH sugerem que somente 5% dos nós sensores necessitam desempenhar o papel de *cluster-head*. Adicionalmente, observou-se que a energia requerida por um nó sensor para transmitir uma mensagem de tamanho unitário a uma distância  $d$  é proporcional a  $d^i$  para  $i$  contido no intervalo  $[2, 4]$  [42] [91] [13]. Portanto, quanto menor a distância entre dois nós sensores  $n_{s_1}$  e  $n_{s_2}$ , menor é consumo de energia [121]. Logo, na versão do LEACH implementada no WSANSim o *cluster-head* não envia diretamente o pacote com os dados agregados para o nó escoadouro. Ao contrário, esse pacote é enviado para o nó ator mais apto, o qual é selecionado pela computação de uma função de custo na qual uma das variáveis avaliada é a menor distância até o *cluster-head*, de modo que esse pode ser empregado como mecanismo de coordenação sensor-sensor e de coordenação sensor-ator no desenvolvimento de protocolos focados no problema da coordenação ator-ator. Contudo, o LEACH apresenta algumas deficiências. Ele não é focado em questões de tempo real. Ademais, o LEACH não pode garantir que o número pré-definido de *cluster-heads* seja uniformemente distribuído entre os nós da rede. Dessa forma, a implementação do LEACH no WSANSim se deve particularmente à disponibilidade do seu código fonte.

- *SinkNodeApplication*: em uma arquitetura de WSN semi-automatizada, o nó escoadouro atua como o elemento central de controle, coletando os dados sensorizados enviados pelos nós sensores e coordenando o processo de atuação por meio do envio de comandos em direção aos nós atores [7]. Portanto, uma instância dessa classe pode ser responsável pelo processo de coordenação dos nós sensores e atores na implementação desse modelo de coordenação.
- *ActorNodeApplication*: instâncias dessa classe estão conectadas em um nó ator e indicam se esse possui energia suficiente para continuar executando ações no ambiente e se comunicar via rede sem fio.

3. Modelo de controle: as classes abaixo são empregadas pelos nós atores para executar

ações no ambiente físico.

- *ControlChannel*: essa classe implementa o comportamento do canal de controle. Ela é responsável pela manutenção das listas de nós atores e de sistemas controlados que estão conectados ao canal de controle, as quais são usadas no mecanismo que decide se um dado nó ator é capaz de executar ações em algum sistema controlado. Um nó ator  $n_{a_i}$  somente pode executar ações em algum sistema controlado  $sc_j$  se todas as condições a seguir são verdadeiras: ele é detentor de um dispositivo de atuação  $d_{a_{ki}}$  que possui a capacidade de se interfacear com algum mecanismo de acoplamento do sistema controlado  $sc_j$ , o tipo de dado sensoriado recebido do nó sensor é compatível com o tipo de fenômeno físico associado a algum mecanismo de acoplamento do sistema controlado  $sc_j$ , se o sistema controlado  $sc_j$  está contido na área de abrangência de um fenômeno físico  $ff_m$  e se existe no mínimo um ponto de intersecção [145] entre o dispositivo de atuação  $d_{a_{ki}}$  do nó ator  $n_{a_i}$  e o sistema controlado  $sc_j$ .
- *Actuator*: essa classe é responsável por implementar o comportamento de um dispositivo de atuação, o qual efetivamente executa uma ação de controle em algum sistema controlado. Um atuador deve estar conectado a alguma instância de nó ator. Adicionalmente, um atuador é proprietário de um módulo de conversão de sinal digital para sinal analógico, o qual converte o sinal de entrada digital gerado pelo processador do nó ator em um sinal de saída analógico que irá ativar o atuador especificado no comando de controle. Um atuador possui as seguintes propriedades: um tipo (o qual indica a que tipo de mecanismo de acoplamento pode o atuador se conectar para a execução de um comando de controle), um raio de atuação (o qual é empregado pelo canal de controle para verificar se um nó ator é capaz de atuar em algum sistema controlado) e um período de tempo para completar a execução do comando de controle.
- *ControlledSystem*: instâncias dessa classe armazenam duas listas, uma que armazena os mecanismos de acoplamento que podem ser empregados para que seja viável a interface de um sistema controlado  $sc_j$  com um dispositivo de atuação  $d_{a_{ki}}$  e outra que indica a qual fenômeno físico  $ff_m$  está associado um dado mecanismo de acoplamento de um sistema controlado  $sc_j$ . Uma instância de sistema controlado possui uma posição geográfica e um raio de controle, os quais são empregados pelo mecanismo de decisão implementado pelo canal de controle.
- *PidController*: um algoritmo de controle PID é um mecanismo de malha fechada genérico amplamente usado em sistemas de controle [15], o qual tenta corrigir o erro entre o valor corrente de uma variável medida do processo (obtida via dispositivo de sensoriamento) e o valor desejado por meio da computação de uma ação corretiva que pode ajustar um dado processo físico em conformidade com a resposta desejada

do sistema.

4. **Modelo de sensoriamento:** é referência comum em muitos dos trabalhos propostos para redes de sensores que cada sensor pode detectar um evento que ocorre dentro do seu raio de sensoriamento, o qual é usualmente assumido como sendo um disco uniforme de raio  $r_s$  [78]. Esse modelo assume que se um evento ocorre a uma distância menor ou igual  $r_s$ , esse pode ser detectado de forma determinística. Se a distância for superior a  $r_s$ , o evento não pode ser detectado. Contudo, diversos trabalhos na literatura têm afirmado que modelos de sensoriamento probabilísticos podem capturar com maior realismo o comportamento dos dispositivos de sensoriamento do que modelo de disco determinístico [184] [102] [185] [4] [36] [78]. Como resultado, constata-se uma tendência crescente em se incorporar modelos probabilísticos aos modelos de propagação de sinal devido ao fato de que esses podem trazer um grau maior de realismo aos ambientes de simulação focados em redes de sensores [122] [159] [48]. A seguir, são descritas as classes empregadas no modelo de propagação de sinal implementado pelo WSANSim.

- *SensorChannel*: analogamente ao que ocorre no SensorSim e no J-Sim, instâncias dessa classe recebem os sinais gerados por instâncias de fenômenos físicos que estão conectados ao canal de sensoriamento e enviam as informações sensoriadas para dispositivos de sensoriamento conectados aos nós sensores que sejam capazes de capturá-las. Nesse contexto, um nó sensor  $n_{s_o}$  pode detectar eventos propagados por um fenômeno físico  $f f_m$  conectado ao canal de sensoriamento se ambas as condições são verdadeiras: o nó sensor possui um dispositivo de sensoriamento  $d_{s_{p_o}}$  cujo o tipo é compatível com o tipo de fenômeno físico  $f f_m$  e a potência do sinal propagado é igual ou superior ao limiar de recepção do dispositivo de sensoriamento. Tal mecanismo depende do modelo de propagação de sinal associado a um fenômeno físico particular.
- *PhysicalPhenomenonPropagationModel*: a partir do emprego da posição geográfica e das propriedades dos fenômenos físicos e dos dispositivos de sensoriamento, um modelo de propagação de sinal pode computar a potência com a qual um dado sinal gerado por um fenômeno físico será recebido, a qual é empregada pelo canal de sensoriamento para transmitir eventos detectados do ambiente para os dispositivos de sensoriamento. Visto que é razoável afirmar que cada tipo de fenômeno físico implica em um modelo de propagação de sinal distinto, novos modelos de propagação de sinal podem ser inseridos no WSANSim.
- *AcousticPropagationModel*: esse modelo de propagação é uma adaptação do modelo de propagação de sinais acústicos implementado no J-Sim [158]. Ele calcula a potência do sinal recebido como uma função da distância entre o emissor e o receptor e dos seguintes parâmetros configuráveis: fator de atenuação do sinal, valor

mínimo, valor máximo, média e variância do ganho de um microfone que apresenta características padronizadas.

- *SeismicPropagationModel*: esse modelo de propagação é uma adaptação do modelo de propagação de sinais sísmicos implementado no J-Sim [158]. Ele calcula a potência do sinal recebido como uma função da distância entre o emissor e o receptor e um fator de atenuação, o qual é um parâmetro configurável.
- *Sensor*: instâncias dessa classe podem detectar eventos em três modalidades: periódica, contínua ou reativa. Um dispositivo de sensoriamento deve ser conectado a um nó sensor. Ele é composto por um módulo de conversão de sinal analógico para sinal digital e por um módulo de processamento de sinais [157]. Um sensor possui um limiar de recepção, o qual é empregado pelos modelos de propagação de sinal para verificar se o dispositivo pode detectar um sinal proveniente do ambiente que foi transmitido para o canal de sensoriamento, e um intervalo de sensoriamento, o qual informa a diferença entre os valores de entrada mínimo e máximo que poderão produzir uma saída válida. Ademais, um sensor possui um tipo (aceleração, força, temperatura, luz, proximidade, etc.), o qual indica que tipo de fenômeno físico pode ser capturado do ambiente.

5. Ambiente: as classes abaixo são empregadas para a representação do comportamento dos objetos do mundo real modelados pelo ambiente de simulação.

- *EnvironmentChannel*: essa classe implementa o canal ambiental. Ela armazena duas listas, uma que contém os sistemas controlados que foram distribuídos no ambiente e outra que contém os fenômenos físicos que podem propagar sinais nesse.
- *PhysicalPhenomenon*: essa classe implementa o comportamento de um fenômeno físico. Esse propaga sinais no canal de sensoriamento em duas modalidades: periódica ou contínua. Uma instância dessa classe deve estar associada a ambos os canais, o ambiental e o de sensoriamento. Uma instância dessa classe deve estar obrigatoriamente associada a uma instância de uma classe que implementa algum modelo de propagação de sinais. Tal composição permite ao canal de sensoriamento escolher os dispositivos de sensoriamento que irão capturar os sinais propagados. Adicionalmente, uma instância de fenômeno físico pode estar associada a uma instância de algum modelo de mobilidade. Um fenômeno físico possui as seguintes propriedades: tipo, posição geográfica, potência de sinal e fator de atenuação (distintos para cada tipo de fenômeno), e um raio de propagação. Essas propriedades são empregadas pelos modelos de propagação de sinais para decidir se um sinal pode ser detectado por algum dispositivo de sensoriamento.
- *MobilityModel*: classe base dos modelos de mobilidade que podem ser conectados aos fenômenos físicos. Acredita-se que o uso de um modelo de mobilidade pode

trazer mais realismo para uma certa categoria de fenômenos físicos que devam ser simulados (por exemplo, na representação de instâncias de focos de incêndio em uma floresta, em um experimento de simulação que requeira o atendimento de tal funcionalidade).

- *RandomWalkMobilityModel*: essa classe implementa o modelo de mobilidade *Random Walk* [39], no qual uma entidade que possui o atributo mobilidade se move da sua posição corrente para uma nova posição por meio da escolha randômica da direção e da velocidade na qual deverá se deslocar. Cada deslocamento ocorre em um intervalo de tempo  $t$  ou distância percorrida  $d$  constantes, ao final do qual uma nova direção e velocidade serão calculadas.
- *RandomWaypointMobilityModel*: essa classe implementa o modelo de mobilidade *Random Waypoint* [88], no qual são incluídos tempos de pausa entre as modificações randômicas da direção e da velocidade a serem percorridas por uma entidade móvel. A cada rodada do mecanismo, essa transita para o estado inicial e após permanece em uma dada localização por um certo período de tempo, e uma vez que esse expire, essa escolhe uma posição destino randômica no terreno de simulação e uma velocidade que é uniformemente distribuída. Após tal evento, a referida entidade se move em direção a nova posição destino na velocidade especificada, e na chegada, espera por um período de tempo antes de começar o processo novamente [39].

6. Modelo de energia: a classe abaixo é usada para representar o consumo de energia dos nós de rede.

- *Battery*: instâncias dessa classe são responsáveis pela redução do nível de energia do nó na execução das operações de sensoriamento, conversão analógico-digital, processamento digital de sinal, conversão digital-analógico, atuação, computação e comunicação. O código dessa classe consiste numa adaptação do código escrito pela equipe do projeto Mannasim.

## 4.5 Estudo de Caso

Com o objetivo de demonstrar o uso do *framework* de simulação proposto, foi criado um cenário de simulação que envolveu a tentativa de se reproduzir o comportamento de um sistema HVAC. Esse tipo de sistema é responsável pelo controle da temperatura e da umidade do ar, pelo fornecimento de ar para a ventilação, pela filtração das partículas do ar e pela movimentação do ar nos espaços ocupados. Acredita-se que esse exemplo simplificado possa ser generalizado para outras aplicações de controle, e provavelmente possa servir como uma prova de conceito para o WSANSim.



### 4.5.1 Modelo de Simulação

Um conjunto de experimentos foram conduzidos com o propósito de avaliar o WSANSim. O modelo de simulação consiste em uma instalação predial composta por quatro salas separadas que possuem propriedades climáticas independentes, de acordo com o que foi reportado no estudo de caso [101]. Em cada sala existe um sensor, sendo que existe somente uma unidade HVAC responsável pelo controle da climatização de todas as salas. Logo, existem quatro nós sensores e somente um nó ator na configuração experimental proposta. No que refere à comunicação entre os nós sensores e o nó ator, foi usado o modelo de comunicação *master/slave*, o qual é frequentemente empregado no âmbito dos protocolos de rede industriais [30] (tais como o MODBUS [129] e o PROFIBUS [130]). Nesse contexto, o nó ator desempenha o papel de mestre e os nós sensores desempenham o papel de escravo. O nó ator envia via *broadcast* uma mensagem de requisição para os nós sensores, enquanto que esses enviam uma mensagem de resposta via *unicast* que contém os dados coletados em cada sala. Após o término de um intervalo de tempo pré-definido, o nó ator executa ações em algum fenômeno físico controlado pelo sistema HVAC através do emprego do algoritmo de controle PID na modalidade proporcional.

O conforto térmico em instalações prediais é afetado por um número de variáveis tais como nível de atividade, vestuário, temperatura, umidade e velocidade do ar [12]. De acordo com a organização ASHRAE (*American Society of Heating, Refrigerating and Air-Conditioning Engineers*), a zona de conforto para a maioria das instalações prediais é definida como o intervalo de condições que produzem 10% de insatisfação térmica. Contudo, tendo em vista a simplificação do modelo, assume-se que todos os fatores são constantes, com exceção da temperatura. A Tabela 2 descreve os principais parâmetros empregados nos experimentos de simulação.

### 4.5.2 Resultados

Com o objetivo de se realizar a testagem do WSANSim, os resultados obtidos nesse experimento foram comparados com os resultados obtidos em um trabalho anterior [101], o qual empregou a tecnologia WSN para realizar o controle de um sistema HVAC (*Heating, Ventilation and Air Conditioning*) [104]. Infelizmente, não foi identificado na literatura nenhum estudo de caso similar que tenha sido implementado em algum ambiente de simulação apresentado na seção 4.2, para que os resultados pudessem ser comparados.

No mencionado trabalho são propostos métodos *ad hoc* e otimizados para o controle de temperatura em múltiplas salas por meio do emprego de uma rede de sensores e de um único atuador. Tal modelo difere das abordagens tradicionais, nas quais um único sensor localizado em uma sala é comumente usado para controlar a temperatura de todas as salas. Tais métodos foram simulados no MATLAB, o qual é uma ferramenta usualmente empregada na simulação

Tabela 2 – Parâmetros de Simulação do Estudo de Caso com o WSANSim

Parâmetro	Valor
Número de experimentos	3 (para cada estratégia de controle)
Tempo de simulação	50 s
Tipo de rede	Heterogênea
Número de nós	5 (4 nós sensores e 1 nó ator)
Área de simulação	20 m x 20 m
Protocolo de acesso ao meio	IEEE 802.11
Protocolo de roteamento	AODV
Protocolo de transporte	UDP
Modelo de propagação de rádio	Two-ray
Mobilidade dos nós	Estacionária
Distribuição dos nós	Uniformemente randômica
Potência de transmissão do rádio	0.28 W (para ambos, nós sensores e o nó ator)
Alcance de comunicação	20 m (sensor), 30 m (ator)
Número de fenômenos físicos	4
Tipo do fenômeno físico propagado	Temperatura
Modelo de mobilidade do fenômenos físicos	Estacionária
Distribuição dos fenômenos físicos no ambiente	Uniformemente randômica
Modelo de propagação de sinal	Sísmico (observação: não foi encontrado na literatura modelo analítico para a temperatura)
Valor médio do sinal propagado	24.87 C° (cidade: Porto Alegre - extraído de: [149])
Desvio padrão do sinal propagado	10% do valor médio
Raio de propagação	30 m
Potência de transmissão do sinal	50000 W (valor usado no J-Sim)
Fator de atenuação do sinal	2
Alcance de sensoriamento	variável (segundo o emprego do modelo de propagação sísmico)
Limiar de recepção do sinal	200 W (valor usado no J-Sim)
Intervalo de amostragem do dispositivo de sensoriamento	1 s
Tipo do dispositivo de sensoriamento	Temperatura
Intervalo de leitura do dispositivo de sensoriamento	-55.0 C° (mínimo), 150.0 C° (máximo) - extraído de: [49]
Intervalo de precisão do dispositivo de sensoriamento	0.975 (mínimo), 1 (máximo) - extraído de: [49]
Raio de atuação	20 m
Tipo do dispositivo de atuação	Eletromagnético
Intervalo de precisão do dispositivo de atuação	0.9 (mínimo), 1 (máximo) - extraído de: [50]
Número de sistemas controlados	1
Raio de controle	20 m
Tipo do mecanismo de acoplamento	Eletromagnético
Valor desejado do sistema controlado	24.5 C° - extraído de: [12]
Modo do controlador PID	Proporcional, $K_p = 1.0$
Energia inicial	10 J (sensor), 100 J (ator)
Energia dispendida na comunicação (recepção)	0.2 W (sensor), 0.43 W (ator)
Energia dispendida na comunicação (transmissão)	0.66 W (sensor), 1 W (ator)
Energia dispendida no sensoriamento	10 mW
Energia dispendida na execução da ação	0.8 W
Energia dispendida no processamento	não considerado

de sistemas de controle [60] [119].

Os resultados dos experimentos simulados em ambos o MATLAB e o WSANSim são apresentados na Tabela 3. Os resultados obtidos no MATLAB indicam que as estratégias de controle *ad hoc* baseadas em múltiplos sensores não são necessariamente melhores do que os resultados obtidos com a estratégia de controle baseada em um único sensor a partir da perspectiva do conforto térmico. Os resultados obtidos com o WSANSim parecem indicar a mesma conclusão. A diferença mais evidente entre os resultados obtidos está relacionado com a estratégia de controle baseada no valor médio das leituras de todos os sensores. Entretanto, é possível afirmar que esse fato se deve à falta de informações sobre a configuração do ambiente de simulação usado no MATLAB.

Tabela 3 – Resultados Experimentais do Estudo de Caso com o WSANSim

Estratégia de controle	Número de salas na zona de conforto ASHRAE	
	MATLAB	WSANSim
Único sensor alocado em uma sala randomicamente escolhida	1.6128	0.9546
Valor médio das leituras de todos os sensores	1.1914	1.9324
Valor médio entre a maior e a menor leitura obtida dos sensores	0.9360	1.0746

## 5 Avaliação de Desempenho

Discutem-se a seguir os resultados obtidos a partir da avaliação de desempenho dos mecanismos de coordenação ator-ator propostos no Capítulo 3. Usualmente, emprega-se a técnica de simulação para tal finalidade. Portanto, a avaliação de desempenho dos mecanismos sugeridos foi conduzida por meio de experimentos de simulação realizados no WSANSim. Particularmente, o presente capítulo busca a identificação de evidências que possam corroborar a efetividade de tais mecanismos.

Pode-se argumentar que da versão base do CNP possa resultar o balanceamento de carga da rede via redistribuição do consumo de energia entre os nós participantes do processo de contratação [156]. Entretanto, visto que tal versão se baseia em um protocolo de comunicação de alto nível que possibilita a negociação entre os nós envolvidos, do qual pode resultar gasto energético significativo [79], sugere-se que a versão base do CNP possa implicar no aumento do tempo de resposta bem como do consumo de energia, e conseqüentemente, possa implicar na diminuição do tempo de vida e da escalabilidade da rede. Inversamente, argumenta-se que a extensão do CNP com o esquema de contratação direta pode implicar no aumento da eficiência do protocolo de negociação [156]. Logo, espera-se que tal extensão possa acarretar na diminuição do tempo de resposta bem como do consumo de energia, e conseqüentemente, possa implicar no aumento do tempo de vida e da escalabilidade da rede.

Similarmente, argumenta-se que o modelo de limiar de resposta apresenta maior escalabilidade e menor tempo de resposta devido ao fato de que esse não necessita de comunicação entre os elementos que compõem o sistema [74]. Porém, da aplicação dos princípios de tal modelo pode resultar um sistema descentralizado que manifesta um comportamento global subótimo e caótico, uma vez que a habilidade de se obter um desempenho global aceitável por meio do uso de estratégias que se baseiam somente em interações locais ainda permanece um desafio [47]. É razoável sugerir que a extensão desse modelo com princípios econômicos pode possibilitar o surgimento de um comportamento global que não é tão caótico quanto o apresentado no modelo de limiar de resposta puro. Sugere-se ainda que a incorporação de informação ambiental em tal modelo possibilita que um dado elemento do sistema adapte suas ações de forma automática a variações provenientes do ambiente [62]. Contudo, é sensato pressupôr que tais acréscimos podem trazer como conseqüência direta o aumento do tempo de resposta do sistema resultante.

## 5.1 Modelo de Simulação

Com o objetivo de avaliar o comportamento dos mecanismos de colaboração propostos, foi criada uma configuração de simulação que possui o objetivo de imitar o comportamento de um sistema de controle do tipo HVAC, o qual é responsável, dentre outras tarefas, pelo controle da temperatura. Mais especificamente, o cenário de simulação elaborado é composto por sensores (que conduzem o sensoriamento da temperatura), atores (que atuam como controladores do sistema HVAC) e fenômenos físicos (os quais simulam a propagação de temperatura para o ambiente).

Os mecanismos de coordenação sensor-sensor e de coordenação sensor-ator adotados no presente modelo de simulação são derivados da implementação do protocolo LEACH no *framework* de simulação WSANSim. Essa é discutida com maior grau de detalhe na seção 4.4. Os parâmetros específicos empregados na simulação do mencionado protocolo de clusterização podem ser observados na Tabela 4. A referida tabela descreve ainda os parâmetros que são genéricos a todos os mecanismos que foram propostos para a coordenação ator-ator. Por sua vez, as Tabelas 5, 6, 7 e 8 descrevem os parâmetros que são específicos a cada mecanismo.

Alguns dos valores dos parâmetros abaixo relacionados puderam ser obtidos de trabalhos similares. Outros foram definidos por meio de procedimento empírico, porquanto não puderam ser extraídos da literatura. Esse ajuste manual naturalmente não é um método muito rigoroso. Porém, a condução de alguns experimentos de simulação com esses valores evidenciaram que tal abordagem conseguiu gerar resultados que pareceram ser satisfatórios. Cabe destacar que abordagem similar foi empregada em proposta que apresentou um modelo de coordenação de tarefas baseado no comportamento colonial de insetos para o contexto de sistemas multi-agentes [47].

Em adição aos parâmetros apresentados na Tabela 5, são acrescentados ao mecanismo de coordenação ator-ator baseado em extensões ao Protocolo de Redes de Contratos os parâmetros visualizados na Tabela 6. Contudo, cabe mencionar que o parâmetro tempo limite para o recebimento de ofertas não é usado na extensão proposta. Por sua vez, o parâmetro  $\delta$  consiste no tempo máximo permitido entre o instante em que um evento é sensoriado do ambiente e o instante no qual a ação associada é completada [108].

Em complemento aos parâmetros apresentados na Tabela 7, acrescenta-se ao mecanismo de coordenação ator-ator baseado em extensões ao Modelo de Limiar e Resposta o parâmetro visualizado na Tabela 8.

Adicionalmente, com a finalidade de se avaliar o efeito da mobilidade dos nós atores sobre o desempenho dos mecanismos propostos, acrescentam-se os parâmetros listados na Tabela 9. Por sua vez, os parâmetros listados na Tabela 10 foram empregados nos experimentos de simulação que empregaram o modelo de propagação de rádio *Shadowing* [133].

Além disso, a Tabela 11 introduz um conjunto de siglas que foram definidas com o intuito

Tabela 4 – Parâmetros Genéricos de Simulação dos Mecanismos de Coordenação Ator-Ator

Parâmetro	Valor
Número de experimentos	33 (extraído de: [141])
Tempo de simulação	100 s (extraído de: [141] [75])
Área de simulação	120 m x 120 m (extraído de: [141])
Número de nós sensores	60
Protocolo de clusterização	LEACH
Número de membros do <i>cluster</i>	Variável
Tempo decorrido entre a eleição dos <i>cluster-heads</i>	10 s
% de nós sensores que atuarão como <i>cluster-heads</i>	5
Intervalo de disseminação de dados sobre um nó ator	5 s
Protocolo de acesso ao meio	IEEE 802.11 (extraído de: [141] [158])
Protocolo de roteamento	AODV (extraído de: [158])
Protocolo de transporte	UDP (extraído de: [141])
Capacidade do nó	50 <i>buffers</i> para a recepção de pacotes (extraído de: [75])
Modelo de mobilidade dos nós sensores	Estacionário (extraído de: [108])
Distribuição dos nós	Uniformemente randômica (extraído de: [141])
Potência de transmissão do rádio	0.85 mW (sensor), 7.21 mW (ator) - extraído de: [90] [162]
Alcance de comunicação	40 m (sensor), 100 m (ator) - extraído de: [75] [108]
Energia inicial	5 J (sensor), uniformemente distribuída entre 800 e 1000 J (ator) - extraído de: [141] [108]
Energia dispendida na comunicação ( <i>reception</i> )	22.20 mW (sensor), 1.2 W (ator) - extraído de: [90] [162]
Energia dispendida na comunicação ( <i>transmission</i> )	22.06 mW (sensor), 1.6 W (ator) - extraído de: [90] [162]
Energia dispendida na comunicação ( <i>idle</i> )	22.06 mW (sensor), 1.0 W (ator) - extraído de: [90] [162]
Energia dispendida na comunicação ( <i>sleep</i> )	9.72 mW (sensor), 0.025 W (ator) - extraído de: [90] [162]
Número de fenômenos físicos	10
Tipo do fenômeno físico propagado	Temperatura
Distribuição dos fenômenos físicos no ambiente	Uniformemente randômica
Modelo de propagação de sinal	Sísmico (observação: não foi encontrado na literatura modelo analítico para a temperatura)
Raio de propagação	Uniformemente distribuído entre 75 e 150 m
Potência de transmissão do sinal	50.000 W (extraído de: [158])
Fator de atenuação do sinal	2 (extraído de: [158])
Alcance de sensoriamento	Variável
Limiar de recepção do sinal	200 W (extraído de: [158])
Intervalo de amostragem do dispositivo de sensoriamento	1 s
Tipo do dispositivo de sensoriamento	Temperatura
Intervalo de leitura do dispositivo de sensoriamento	-55.0 C° (mínimo), 150.0 C° (máximo) - extraído de: [49]
Intervalo de precisão do dispositivo de sensoriamento	0.975 (mínimo), 1 (máximo) - extraído de: [49]
Energia dispendida no sensoriamento	10 mW (extraído de: [141])
Número de sistemas controlados	10
Raio de controle	30 m
Tipo do mecanismo de acoplamento	Eletromagnético
Raio de atuação	30 m (extraído de: [175] [75])
Tipo do dispositivo de atuação	Eletromagnético
Tempo que o dispositivo leva para completar a ação	1 s (extraído de: [50])
Intervalo de precisão do dispositivo de atuação	0.9 (mínimo), 1 (máximo) - extraído de: [50]
Energia dispendida na execução da ação	5.6 W (extraído de: [50])
Energia dispendida no processamento	não considerado

Tabela 5 – Parâmetros Específicos de Simulação - Coordenação Ator-Ator Baseada no Protocolo de Redes de Contratos

Parâmetro	Valor
Tempo limite para o recebimento de lances	10 s
$\alpha$	3.0
$\beta$	3.0
$\gamma$	4.0
$\lambda$	1.0
$\delta$	25 s

Tabela 6 – Parâmetros Específicos de Simulação - Coordenação Ator-Ator Baseada em Extensões ao Protocolo de Redes de Contratos

Parâmetro	Valor
Intervalo de disseminação de mensagens do tipo <i>node available</i>	5 s
Tempo de expiração de uma mensagem do tipo <i>node available</i>	10 s

Tabela 7 – Parâmetros Específicos de Simulação - Coordenação Ator-Ator Baseada no Modelo de Limiar de Resposta

Parâmetro	Valor
Modelo de limiar de resposta	Fixo (extraído de: [24] [74])
$\theta$	20.0 (extraído de: [24])
$\omega$	1.0
$\alpha$	50.0
$\beta$	50.0
$\gamma$	50.0
$\lambda$	10.0
$\rho$	50.0
$\delta$	25 s
$n$	2 (extraído de: [24])

Tabela 8 – Parâmetros Específicos de Simulação - Coordenação Ator-Ator Baseada em Extensões ao Modelo de Limiar de Resposta

Parâmetro	Valor
Tempo limite para a avaliação dos lances	10 s

Tabela 9 – Parâmetros Específicos de Simulação - Modelo de Mobilidade Empregado pelos Nós Atores

Parâmetro	Valor
Modelo de mobilidade	<i>Random Walk</i>
Intervalo de atualização da posição do nó	5 s
Velocidade máxima	10 m/s
Energia dispendida na movimentação do nó	não considerado

Tabela 10 – Parâmetros Específicos de Simulação - Modelo de Propagação de Rádio *Shadowing*

Parâmetro	Valor
<i>Path loss exponent</i>	4
<i>Shadowing deviation</i>	6.8 dB

de facilitar a discussão dos resultados obtidos nos experimentos de simulação.

Tabela 11 – Siglas Empregadas na Identificação dos Mecanismos Propostos para a Coordenação Ator-Ator

Mecanismo	Sigla
Baseado no Protocolo de Redes de Contratos	BPRC
Baseado em Extensões ao Protocolo de Redes de Contratos	BEPRC
Baseado no Modelo de Limiar de Resposta	BMLR
Baseado em Extensões ao Modelo de Limiar de Resposta	BEMLR

## 5.2 Requisitos Avaliados

Uma rede de sensores e atores sem fio consiste num sistema de controle distribuído cooperativo que deve reagir a estímulos provenientes do ambiente de acordo com restrições temporais. Visto que tal sistema é composto por nós que possuem características heterogêneas, e que assume-se nesse trabalho uma WSN que possui uma arquitetura automatizada, surge nesse contexto o fenômeno da coordenação. Além do mais, a eficiência energética é fundamental, já que assume-se que a energia necessária para atuar no ambiente é maior do que a energia necessária para a comunicação. Por isso, é importante que se minimize o consumo de energia na etapa de atuação com o objetivo de se prolongar o tempo de vida da rede. Portanto, a seguir verifica-se se tais requisitos elencados particularmente para o contexto da coordenação ator-ator foram atendidos pelos protocolos que foram elaborados para a concretização de tal coordenação.

## 5.3 Medidas de Desempenho

Nesse capítulo, procura-se avaliar o desempenho dos protocolos propostos para a coordenação ator-ator por meio do emprego das medidas abaixo relacionadas:

- **Tempo de resposta do sistema:** tempo decorrido entre o instante em que um evento é sensoriado do ambiente e o instante no qual a ação associada é completada.
- **Taxa de sucesso no atendimento da restrição temporal:** se refere ao percentual de tarefas que foram executadas dentro do  $\delta$  definido no *setup* experimental.
- **Taxa de sucesso na cobertura das áreas de eventos:** se refere à divisão do número total de vezes que ações foram executadas e que atenderam ao critério de cobertura geográfica das áreas de eventos pelo número total de vezes que tais ações foram executadas.



- **Quantidade de energia residual do conjunto de nós atores:** se refere à computação da quantidade de energia residual média do número total de nós atores presentes em cada repetição do experimento de simulação.

Acredita-se que essas medidas possam auxiliar na caracterização da habilidade dos protocolos propostos em satisfazerem os requisitos intrínsecos da coordenação ator-ator [7]. Ademais, avalia-se o grau de escalabilidade de cada mecanismo proposto.

## 5.4 Resultados Experimentais

A seguir são discutidos os resultados que foram obtidos a partir da condução dos experimentos de simulação. É de fundamental importância, analisa-se se tais resultados conseguem demonstrar a efetividade dos mecanismos propostos no que se refere ao atendimento dos requisitos anteriormente elucidados. Cabe enfatizar que algumas das medidas de desempenho foram obtidas a partir da simples mensuração de uma taxa de ocorrência, enquanto que outras foram obtidas pela análise estatística dos dados gerados pelos experimentos de simulação. Nesse caso específico, mediu-se a média ( $\mu$ ), o desvio padrão ( $\sigma$ ) e o coeficiente de variação ( $cv = \frac{\sigma}{\mu}$ ). Cabe destacar que o último indicador permite a comparação do grau de variabilidade do comportamento de dois conjuntos de dados [92].

Cabe enfatizar que o *deadline* ( $\delta$ ) parametrizado para todos os mecanismos é de 25 segundos, cujo valor foi obtido de [175]. Em trabalho similar [108], o  $\delta$  assumido é de 10 segundos. A partir de um conjunto de simulações iniciais, constatou-se que os protocolos propostos para a coordenação ator-ator não conseguiam satisfazer essa mesma restrição, a qual sugere-se que foi atendida no trabalho relacionado. Devido a esse fato, conduziram-se explorações empíricas que tentaram constatar o motivo. Esse foi localizado na implementação do protocolo LEACH no WSANSim, o qual introduziu um *overhead* médio de 12,27 segundos nos mecanismos de coordenação ator-ator elaborados.

Uma análise apropriada de uma instância de projeto experimental pode auxiliar na identificação dos fatores que podem afetar o desempenho de um dado sistema computacional [86]. Um fator é uma variável que afeta os resultados obtidos de um experimento e possui níveis, os quais são os valores que o fator pode assumir. Mais especificamente, uma instância de projeto experimental consiste na especificação dos seguintes elementos: o número de experimentos, as combinações dos níveis dos fatores e o número de repetições de cada experimento. Existem vários tipos de projeto experimental. De particular importância, o fatorial fracionado [86], no qual assume-se uma configuração típica de parâmetros e varia-se uma fração dos fatores que possam afetar o desempenho do sistema em questão. Essa é a modalidade de projeto empregada na avaliação de desempenho dos mecanismos de coordenação ator-ator propostos na presente dissertação.

Na avaliação de desempenho que se segue foram escolhidos dois fatores que podem implicar em efeitos significativos no desempenho dos mecanismos de colaboração ator-ator elaborados no que se refere à comunicação sem fio entre os nós. O primeiro consiste no modelo de mobilidade adotado pelos nós atores, enquanto que o segundo consiste no modelo de propagação de sinal de rádio empregado pelos nós sensores e pelos nós atores. Tal escolha se deve estritamente a dificuldades operacionais que foram encontradas na variação de outros fatores que poderiam trazer impactos diretos no desempenho dos mecanismos propostos tais como o protocolo de acesso ao meio, o protocolo de roteamento e o protocolo de transporte. Outro fator que foi avaliado foi o efeito do número de nós atores sobre o desempenho dos mecanismos de colaboração propostos. Particularmente, foram elaborados cenários de simulação que empregaram 12 nós atores (5 nós sensores para cada nó ator), 15 nós atores (4 nós sensores para cada nó ator), 20 nós atores (3 nós sensores para cada nó ator) e 30 nós atores (2 nós sensores para cada nó ator).

#### 5.4.1 Efeito do Modelo de Mobilidade

O conceito de ator móvel é relativamente novo na área de redes de sensores e atores sem fio [96]. A principal vantagem auferida por um ator móvel consiste no fato de que ações controle podem ser executadas de forma mais efetiva se o ator estiver próximo da área na qual está localizado o evento sensoriado. Um dos modelos de mobilidade comumente adotados para a avaliação do efeito desses sobre o desempenho de propostas de pesquisa na área de WSN é o modelo *Random Walk*, ou variações desse modelo. Esse modelo foi proposto originalmente para imitar o movimento randômico e não determinístico de partículas subatômicas no campo da área da Física Nuclear, e é o modelo de mobilidade empregado pelos nós atores nos cenários que simulam o movimento desses. Modelos de mobilidade randômicos são modelos não inteligentes [96]. Contudo, o seu uso na avaliação de desempenho de algoritmos e protocolos focados em redes WSN é ainda predominante. Logo, discutem-se a seguir os efeitos produzidos pelo uso de atores estáticos e pelo uso de móveis móveis sobre os quatro mecanismos de colaboração ator-ator propostos. Cabe ressaltar que em ambos os cenários de simulação é usado o modelo de propagação de rádio *Shadowing*, o qual simula com maior realismo o comportamento do nível físico de um canal de rede sem fio [133].

A influência do uso do modelo de mobilidade estático e do número variável de atores sobre os índices de desempenho dos mecanismos de coordenação ator-ator propostos é explanado a seguir, a qual é quantificada pelos resultados apresentados na Figura 20.

A partir dos resultados contidos na Figura 20(a), verifica-se que o tempo de resposta médio do BPRC é superior ao tempo de resposta médio do BEPRC, o que se deve ao fato de que o BEPRC emprega um mecanismo de contratação simplificado. Tais resultados evidenciam que o tempo de resposta médio do BPRC e do BEPRC não diminui de forma significativa com o aumento do número de atores. Logo, tais mecanismos não são escaláveis no que se refere a essa

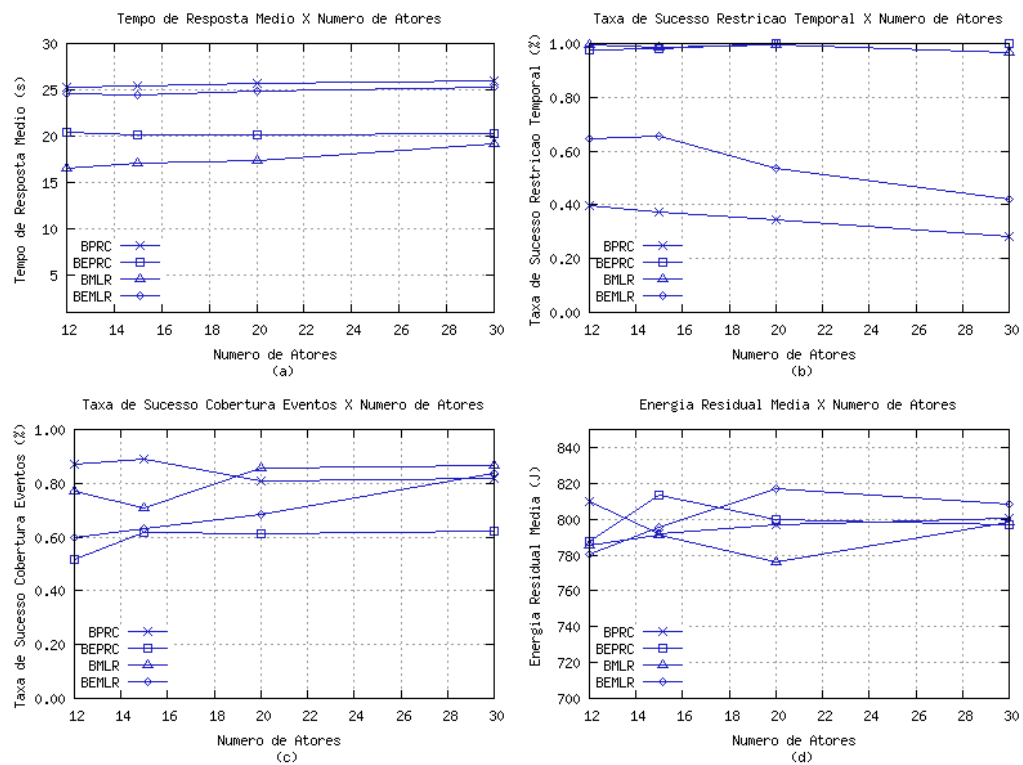


Figura 20 – Desempenho dos Mecanismos de Coordenação Ator-Ator para o Modelo de Mobilidade Estático em Redes com Diferentes Números de Atores (a) Tempo de Resposta Médio (b) Taxa de Sucesso no Atendimento de Limitações Temporais (c) Taxa de Sucesso na Cobertura das Áreas de Eventos (d) Energia Residual Média

medida. Quanto ao BMLR e ao BEMLR, constata-se que o tempo de resposta médio do BEMLR é superior ao tempo de resposta médio do BMLR, o que se deve ao fato de que no BMLR não se faz necessária a comunicação entre os atores, pois que esses decidem de forma independente se devem executar ações no ambiente controlado. Complementarmente, os resultados obtidos pelo BMLR sugerem que o tempo de resposta médio aumenta quando ocorre o aumento do número de atores. Por sua vez, os resultados obtidos pelo BEMLR evidenciam que o tempo de resposta médio desse mecanismo não é afetado de forma significativa pelo aumento do número de atores. Portanto, esses resultados sugerem que ambos os mecanismos não são escaláveis no que se refere à referida medida de desempenho.

Tal tendência se reflete nos resultados apresentados na Figura 20(b). Esses indicam que a taxa de sucesso no atendimento de limitações temporais do BEPRC é maior do que a taxa de sucesso do BPRC. Tais resultados indicam ainda que a taxa de sucesso no atendimento de limitações temporais do BEPRC aumenta de forma levemente significativa com o aumento do número de atores. Contudo, as evidências experimentais indicam que tal afirmação não é verdadeira para o BPRC, cuja a taxa diminui de forma expressiva com o aumento do número de nós atores. Tais resultados sugerem que o BPRC não é escalável, enquanto que sugere-se que o BEPRC possa a vir manifestar a propriedade de escalabilidade. No que se refere ao BMLR e ao BEMLR, os resultados indicam que a taxa de sucesso no atendimento de limitações temporais do BMLR é maior do que a taxa de sucesso do BEMLR. Tais resultados evidenciam ainda que no BMLR tal medida diminui de forma menos significativa quando o número de atores varia, enquanto que no BEMLR tal índice de desempenho diminui de forma bem mais considerável. Dessa forma, sugere-se que ambos os mecanismos não sejam escaláveis no que se refere à taxa de sucesso no atendimento de limitações temporais.

De acordo com os resultados apresentados na Figura 20(c), observa-se que a taxa de sucesso na cobertura das áreas de eventos do BPRC é maior do que a taxa de sucesso do BEPRC. Tal comportamento encontra justificativa no fato de que o BEPRC pode selecionar o ator ou o conjunto de atores mais aptos a partir de informações disponibilizadas sobre o seu estado passado, de tal forma que o gerente pode delegar a execução de um contrato dirigido a um ator que não é capaz de satisfazer à especificação de elegibilidade do contrato corrente. Tais resultados sugerem que o desempenho do BPRC diminui com o aumento do número de atores, enquanto que no BEPRC o desempenho aumenta de forma levemente significativa de acordo com o aumento do número desses. Contudo, tais efeitos não puderam ser identificados de forma muito precisa nos experimentos que foram efetuados. Quanto aos resultados obtidos pelo BMLR e pelo BEMLR, constata-se que a taxa de sucesso na cobertura das áreas de eventos do BMLR é maior do que a taxa de sucesso do BEMLR. Adicionalmente, tais resultados indicam que o desempenho do BEMLR aumenta de acordo com o aumento do número de atores, enquanto que tal tendência não pôde ser detectada com certo grau de precisão no BMLR. Portanto, sugere-se que o BEMLR seja escalável no que se refere à referida medida de desempenho.

Conforme os resultados visualizados na Figura 20(d), constata-se que a energia residual

Tabela 12 – Medidas de Variabilidade da Quantidade de Energia Residual do Conjunto de Nós Atores para os Mecanismos de Coordenação Ator-Ator Baseados no Modelo de Mobilidade Estático em Redes com Diferentes Números de Atores

Mecanismo	12 atores	15 atores (s)	20 atores	30 atores
BPRC	$\sigma (J) = 64,1444$ $cv = 0,0792$	$\sigma (J) = 57,8157$ $cv = 0,0730$	$\sigma (J) = 51,6762$ $cv = 0,0648$	$\sigma (J) = 58,9500$ $cv = 0,0737$
BEPRC	$\sigma (J) = 51,7677$ $cv = 0,0658$	$\sigma (J) = 54,7565$ $cv = 0,0673$	$\sigma (J) = 64,1799$ $cv = 0,0803$	$\sigma (J) = 62,7265$ $cv = 0,0787$
BMLR	$\sigma (J) = 61,7634$ $cv = 0,0786$	$\sigma (J) = 49,2421$ $cv = 0,0622$	$\sigma (J) = 49,5181$ $cv = 0,0638$	$\sigma (J) = 42,6988$ $cv = 0,0535$
BEMLR	$\sigma (J) = 56,6608$ $cv = 0,0726$	$\sigma (J) = 59,3654$ $cv = 0,0746$	$\sigma (J) = 57,1831$ $cv = 0,0700$	$\sigma (J) = 54,2793$ $cv = 0,0671$

média do conjunto de nós atores possui um certo nível de similaridade tanto no BPRC quanto no BEPRC, quando o número de atores é igual a 20 ou 30. Tal padrão se reproduz nos valores dos coeficientes de variação calculados para ambos os mecanismos de colaboração (ver Tabela 12), os quais possuem um certo grau de semelhança. Além disso, os resultados obtidos por tais mecanismos sugerem que o dispêndio energético seja redistribuído de forma uniforme entre os atores de acordo com o aumento do número desses, o que pode implicar no aumento do tempo de vida da rede. Exceção ocorre no cenário de simulação no qual o número de atores é igual 15, no qual verifica-se que a energia residual média do BEPRC supera a energia residual média do BPRC. Essa é ainda constatada quando o número de atores é igual 12, na qual observa-se que a energia residual média do BPRC supera a energia residual média do BEPRC. No que se refere ao BMLR e ao BEMLR, os resultados evidenciam que a energia residual média do conjunto de atores no BEMLR é maior do que no BMLR. Tal resultado se deve ao fato de que no BMLR a decisão de executar uma ação de controle sobre um dado sistema controlado é tomada de forma independente e descentralizada por cada ator, ao contrário do que ocorre no BEMLR, o que implica no aumento do consumo de energia. Quanto ao efeito do número de atores sobre a energia residual média do conjunto de atores em ambos os mecanismos, esse não pôde ser quantificado de maneira precisa pelos experimentos que foram conduzidos.

Por sua vez, o impacto do emprego do modelo de mobilidade *Random Walk* e do número variável de atores sobre os índices de desempenho dos mecanismos de coordenação ator-ator elaborados é examinado a seguir, o qual é mensurado pelos resultados apresentados na Figura 21.

A partir dos resultados contidos na Figura 21(a), verifica-se que o tempo de resposta médio do BPRC é superior ao tempo de resposta médio do BEPRC, e que o tempo de resposta médio do BPRC e do BEPRC não diminui de forma significativa com o aumento do número de atores. Logo, sugere-se que ambos os mecanismos não sejam escaláveis no que se refere a esse índice. Quanto ao BMLR e ao BEMLR, constata-se que o tempo de resposta médio do BEMLR é superior ao tempo de resposta médio do BMLR. Os resultados obtidos pelo BMLR sugerem que o tempo de resposta médio desse aumenta quando ocorre o aumento do número de atores, enquanto que os resultados obtidos pelo BEMLR evidenciam que o tempo de resposta médio

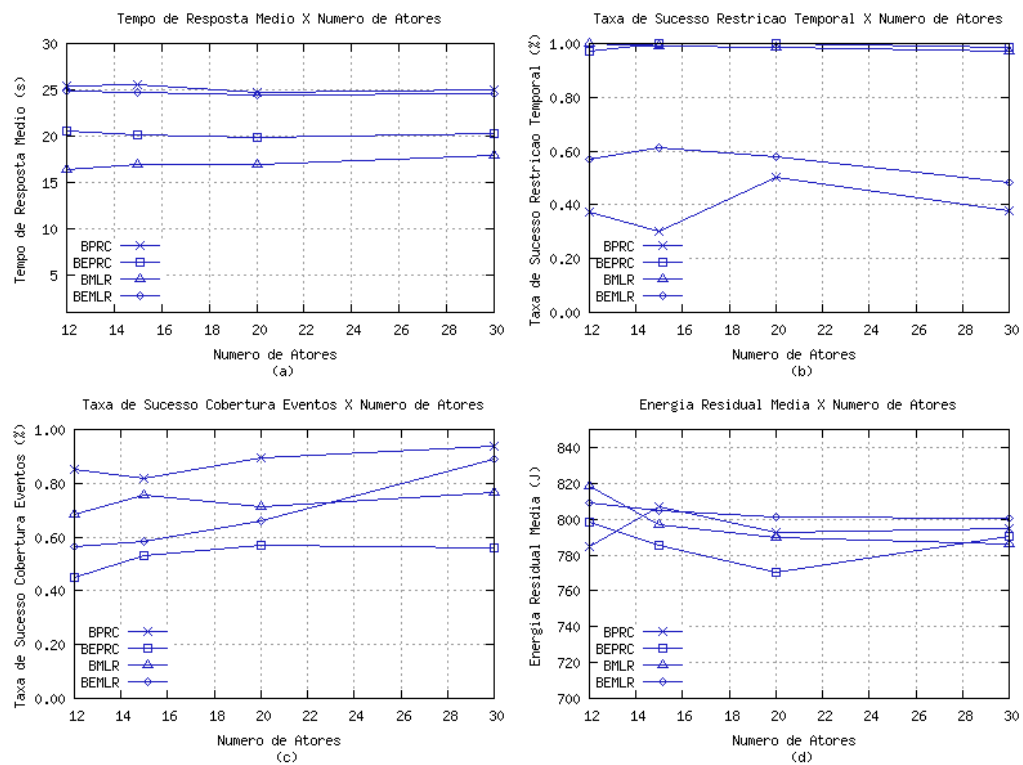


Figura 21 – Desempenho dos Mecanismos de Coordenação Ator-Ator para o Modelo de Mobilidade *Random Walk* em Redes com Diferentes Números de Atores (a) Tempo de Resposta Médio (b) Taxa de Sucesso no Atendimento de Limitações Temporais (c) Taxa de Sucesso na Cobertura das Áreas de Eventos (d) Energia Residual Média

desse mecanismo não é afetado de forma significativa pelo aumento do número de atores. Portanto, ambos não são escaláveis no que se refere a essa medida. Tais resultados são similares aos que foram obtidos por meio do emprego do modelo de mobilidade estático. Logo, sugere-se que o uso de atores que se movem segundo o modelo de mobilidade *Random Walk* bem como o uso de um número variável de atores não afeta o desempenho dos mecanismos propostos no que se refere ao tempo de resposta médio.

Tal comportamento se reproduz nos resultados apresentados na Figura 21(b). Esses indicam que a taxa de sucesso no atendimento de limitações temporais do BEPRC é maior do que a taxa de sucesso do BPRC, e que a taxa de sucesso no atendimento de limitações temporais do BEPRC se mantém levemente inalterada conforme ocorre o aumento do número de atores. Entretanto, os dados experimentais sugerem que tal afirmação não é verdadeira para o BPRC, cuja a taxa diminui de acordo com o aumento do número de nós atores. Dessa forma, tais mecanismos não são escaláveis no que se refere a essa taxa. No que se refere ao BMLR e ao BEMLR, os resultados indicam que a taxa de sucesso no atendimento de limitações temporais do BMLR é maior do que a taxa de sucesso do BEMLR. Tais resultados evidenciam ainda que no BMLR tal medida diminui de forma menos significativa quando o número de atores varia, enquanto que no BEMLR tal índice de desempenho diminui de forma bem mais considerável. Conseqüentemente, o BMLR e o BEMLR não apresentam a propriedade de escalabilidade. Cabe ressaltar que tais resultados são similares aos que foram obtidos pelo emprego do modelo de mobilidade estático. Portanto, sugere-se que o emprego de um número variável de atores que se movem segundo o modelo de mobilidade *Random Walk* não produz efeitos significativos sobre o desempenho dos mecanismos elaborados no que se refere à taxa de sucesso no atendimento de limitações temporais.

Segundo os resultados apresentados na Figura 21(c), observa-se que a taxa de sucesso na cobertura das áreas de eventos do BPRC é maior do que a taxa de sucesso do BEPRC. Tais resultados sugerem que o desempenho do BPRC aumenta de acordo com o aumento do número de atores, ao contrário do que foi constatado no cenário de simulação anterior. Quanto ao BEPRC, o desempenho aumenta de acordo com o aumento do número desses, o que já havia sido constatado no cenário de mobilidade estático. Portanto, tais resultados indicam que ambos os mecanismos são escaláveis no que tange a essa medida. Quanto aos resultados obtidos pelo BMLR e pelo BEMLR, constata-se que a taxa de sucesso na cobertura das áreas de eventos do BMLR é maior do que a taxa de sucesso do BEMLR enquanto o número de atores é igual a 20. Quando número de atores é superior a esse número, os resultados evidenciam que o desempenho do BEMLR supera o desempenho do BMLR. Adicionalmente, os resultados indicam que o desempenho do BEMLR aumenta de acordo com o aumento do número de atores, ao contrário do que o ocorre com o BMLR. Logo, os resultados indicam que o BEMLR possui a propriedade de escalabilidade. Dessa forma, sugere-se que o emprego de um número variável de atores que se movem segundo o modelo de mobilidade *Random Walk* produz efeitos consideráveis sobre o desempenho dos mecanismos desenvolvidos no que se refere à taxa de sucesso na cobertura

das áreas de eventos. Tal resultado era esperado, uma vez que um nó ator móvel pode executar ações controle de forma mais efetiva se esse estiver próximo da área na qual está localizado o evento que foi detectado pelos nós sensores.

Tabela 13 – Medidas de Variabilidade da Quantidade de Energia Residual do Conjunto de Nós Atores para os Mecanismos de Coordenação Ator-Ator Baseados no Modelo de Mobilidade *Random Walk* em Redes com Diferentes Números de Atores

Mecanismo	12 atores	15 atores (s)	20 atores	30 atores
BPRC	$\sigma (J) = 50,1748$ $cv = 0,0639$	$\sigma (J) = 58,2601$ $cv = 0,0722$	$\sigma (J) = 59,6980$ $cv = 0,0753$	$\sigma (J) = 55,7692$ $cv = 0,0702$
BEPRC	$\sigma (J) = 64,5601$ $cv = 0,0808$	$\sigma (J) = 52,0477$ $cv = 0,0663$	$\sigma (J) = 53,9432$ $cv = 0,0700$	$\sigma (J) = 53,8474$ $cv = 0,0681$
BMLR	$\sigma (J) = 54,7206$ $cv = 0,0669$	$\sigma (J) = 56,1339$ $cv = 0,0704$	$\sigma (J) = 52,3324$ $cv = 0,0663$	$\sigma (J) = 49,9804$ $cv = 0,0636$
BEMLR	$\sigma (J) = 50,2628$ $cv = 0,0621$	$\sigma (J) = 58,0552$ $cv = 0,0721$	$\sigma (J) = 44,6516$ $cv = 0,0557$	$\sigma (J) = 59,8118$ $cv = 0,0747$

De acordo com os resultados apresentados na Figura 21(d), observa-se que a energia residual média do conjunto de nós atores quando o número de atores é igual a 12 no BEPRC é superior a energia residual média do BPRC. Todavia, a energia residual média do BPRC supera a energia residual média do BEPRC quando o número de atores é superior a 12. Esse grau de variabilidade pode ser constatado de forma mais precisa nos valores assumidos pelos coeficientes de variação de ambos os mecanismos para o cenário de simulação que emprega 12 atores, os quais são apresentados na Tabela 13. Quanto aos efeitos produzidos pelo número variável de atores sobre o desempenho dos referidos mecanismos, sugere-se que o desempenho energético de ambos diminui pelo aumento do número de atores, o que indica que esses não possuem escalabilidade energética. Contudo, os resultados experimentais não permitem a identificação de tal tendência de forma tão precisa. No que se refere ao BMLR e ao BEMLR, os resultados indicam que a energia residual média do conjunto de atores no BMLR é maior do que no BEMLR quando o número de atores é igual a 12. Contudo, a energia residual média do BEMLR ultrapassa a energia residual média do BMLR quando o número de atores é superior a 12 (comportamento similar já havia sido identificado no caso do BPRC e do BEPRC). Quanto ao efeito do número de atores sobre o desempenho desses mecanismos, verifica-se que a energia residual média do BMLR diminui de forma considerável conforme ocorre o aumento do número de atores, enquanto que a energia residual média do BEMLR diminui de forma menos considerável. Entretanto, constata-se a partir dos resultados apresentados na Tabela 13 para o BMLR que os coeficientes de variação do BMLR assumem valores que possuem um maior nível de homogeneidade entre si. Tal comportamento pode implicar no balanceamento do consumo energético, o que pode acarretar no acréscimo do tempo de vida da rede. Similarmente ao que foi constatado no caso do BMLR por meio das estatísticas obtidas para esse, os resultados visualizados na Figura 21(d) sugerem que o BEMLR consegue redistribuir o consumo de energia de forma uniforme, o que pode implicar no aumento do tempo de vida do sistema. Entretanto, tais resultados em sua totalidade indicam que ambos os mecanismos não possuem escalabili-



dade energética. Dessa forma, sugere-se que o emprego de um número variável atores que se movem segundo o modelo de mobilidade *Random Walk* afeta o desempenho dos mecanismos projetados no que se refere à energia residual média do conjunto de nós atores.

#### 5.4.2 Efeito do Modelo de Propagação de Rádio

Os modelos de propagação de rádio são empregados para que possa prever a potência do sinal de cada pacote recebido do canal de rede sem fio [128]. No caso específico das extensões produzidas pelo projeto Monarch, na camada física de cada nó móvel existe um limiar de recepção. Esse é calculado de forma distinta para cada modelo de propagação de rádio disponibilizado nas bibliotecas do ns-2, os quais se utilizam dentre outros parâmetros particulares para cada modelo, do alcance de comunicação entre os nós móveis. Quando um pacote é recebido do canal de rede sem fio, ele é descartado pela camada de acesso ao meio se a potência do sinal recebido é inferior ao limiar de recepção. Em caso contrário, esse é enviado para as camadas de rede superiores.

Usualmente, empregam-se dois modelos de propagação de rádio para a avaliação do efeito desses sobre o desempenho de trabalhos propostos na áreas de WSN e WSAN: o modelo *Free space* e o modelo *Shadowing*. O primeiro assume a condição ideal de propagação na qual existe um caminho de propagação direto entre o transmissor e o receptor [133]. Esse modelo representa o alcance de comunicação como um disco de raio uniforme  $r$ , de modo que se um receptor  $j$  está dentro desse círculo ideal, ele recebe todos os pacotes que foram enviados pelo transmissor  $i$ . Em caso contrário, todos os pacotes recebidos por  $j$  de  $i$  são descartados. Como resultado, esse modelo realiza a predição da potência do sinal recebido como uma função determinística da distância entre os nós comunicantes. Ao contrário daquele, o segundo modelo assume que a potência do sinal recebido em uma dada distância é uma variável randômica devido aos efeitos dos fenômenos da interferência, da reflexão e da refração do sinal no meio de propagação [133]. Dessa forma, o presente modelo afirma que os nós podem se comunicar somente de forma probabilística quando existe uma relação de proximidade entre os alcances de comunicação dos nós transmissor e receptor, o qual modela de forma mais realística o fenômeno da propagação de sinais de rádio no canal de rede sem fio. A seguir, verificam-se quais são os efeitos produzidos pelo emprego de tais modelos de propagação sobre os quatro mecanismos de colaboração ator-ator propostos. Cabe mencionar que em ambos os cenários de simulação é empregado o modelo de mobilidade estático.

Os efeitos produzidos pelo modelo de propagação de rádio *Free Space* e pelo número variável de atores sobre os índices de desempenho dos mecanismos de coordenação ator-ator projetados são discutidos abaixo, os quais são quantificados pelos resultados apresentados na Figura 22.

Conforme os resultados contidos na Figura 22(a), observa-se que o tempo de resposta médio

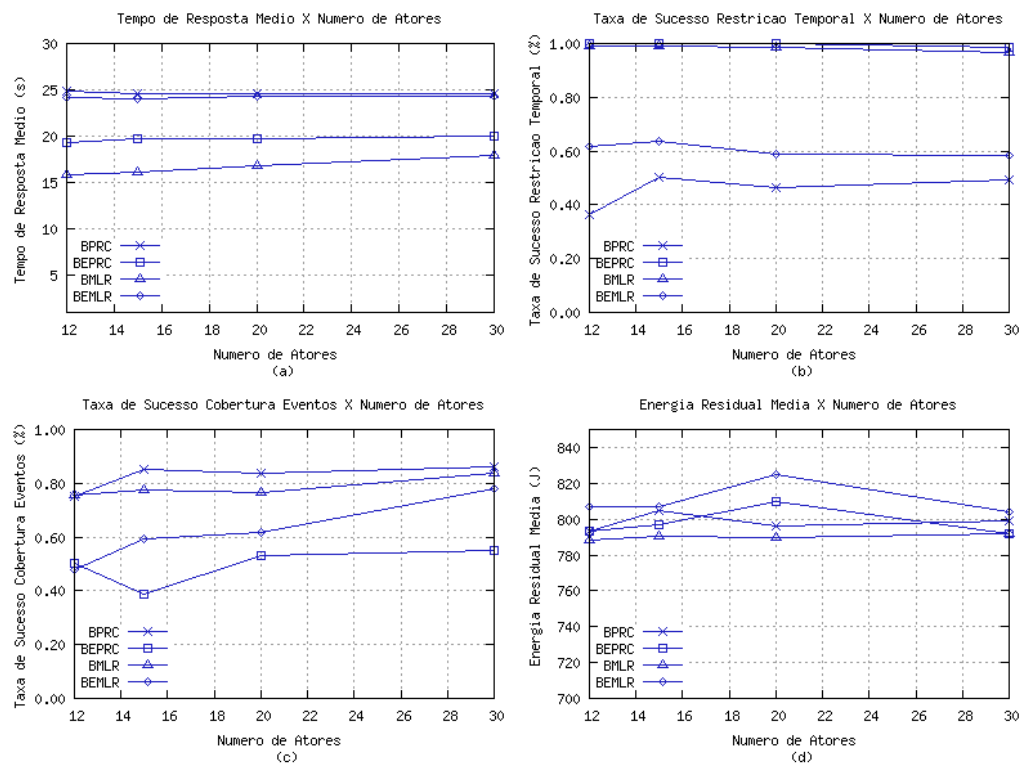


Figura 22 – Desempenho dos Mecanismos de Coordenação Ator-Ator para o Modelo de Propagação de Rádio *Free Space* em Redes com Diferentes Números de Atores (a) Tempo de Resposta Médio (b) Taxa de Sucesso no Atendimento de Limitações Temporais (c) Taxa de Sucesso na Cobertura das Áreas de Eventos (d) Energia Residual Média

do BPRC é superior ao tempo de resposta médio do BEPRC. Esses resultados indicam ainda que o tempo de resposta médio do BPRC e do BEPRC não diminui de forma significativa com o aumento do número de atores. Portanto, não são escaláveis a respeito dessa medida. Quanto aos resultados obtidos pelo BMLR e pelo BEMLR, verifica-se que o tempo de resposta médio do BEMLR é superior ao tempo de resposta médio do BMLR. Tais resultados sugerem que o tempo de resposta médio do BMLR aumenta quando ocorre o aumento do número de atores, à medida que o tempo de resposta médio do BEMLR não é afetado de forma significativa pelo aumento do número de atores. Por conseguinte, ambos os mecanismos não são escaláveis no que se refere a essa medida.

Tal tendência pode ser verificada nos resultados apresentados na Figura 22(b). Esses evidenciam que a taxa de sucesso no atendimento de limitações temporais do BEPRC é maior do que a taxa de sucesso do BPRC, e que a referida taxa do BEPRC não varia de forma significativa com o aumento do número de atores. Quanto ao BPRC, constata-se que a essa taxa varia de forma mais significativa conforme ocorre o aumento do número de nós atores. Entretanto, tais resultados indicam que ambos os mecanismos não sejam escaláveis no que se relaciona a essa medida. No que se refere ao BMLR e ao BEMLR, os resultados apontam que a taxa de sucesso no atendimento de limitações temporais do BMLR é maior do que a taxa de sucesso do BEMLR. Tais resultados sugerem ainda que o referido índice de desempenho diminui de forma levemente significativa tanto no BMLR quanto no BEMLR quando o número de atores varia. Portanto, ambos não são escaláveis no que se refere a essa taxa.

De acordo com os resultados apresentados na Figura 22(c), constata-se que a taxa de sucesso na cobertura das áreas de eventos do BPRC é maior do que a taxa de sucesso do BEPRC. Esses sugerem ainda que o desempenho do BPRC aumenta de forma pouco considerável de acordo com o aumento do número de atores, enquanto que no BEPRC o desempenho diminui quando o número de atores varia de 12 para 15, e aumenta quando esse número é superior a 15. Dessa maneira, os resultados não permitem que sejam tecidas conclusões a respeito da escalabilidade de tais mecanismos. Quanto aos resultados obtidos pelo BMLR e pelo BEMLR, constata-se que a taxa de sucesso na cobertura das áreas de eventos do BMLR é maior do que a taxa de sucesso do BEMLR. Complementarmente, tais resultados indicam que o desempenho tanto do BMLR aumenta levemente conforme ocorre o aumento do número de atores. No caso específico do BEMLR, as evidências indicam que o desempenho desse aumenta de forma mais significativa de acordo com o aumento do número de atores. Logo, sugere-se que o BMLR e o BEMLR possam vir a manifestar a propriedade de escalabilidade no que refere a essa medida.

Segundo os resultados apresentados na Figura 22(d), constata-se que a energia residual média do conjunto de nós atores do BEPRC é superior a energia residual média do BPRC quando o número de atores é igual a 20, enquanto que nos outros cenários (12, 15 e 30 atores) a energia residual média do BPRC é igual ou superior a energia residual média do BEPRC. Tal variabilidade pode ser constatada nos valores assumidos pelas medidas de dispersão (desvio padrão e coeficiente de variação) computados para ambos os mecanismos para os diferentes números

Tabela 14 – Medidas de Variabilidade da Quantidade de Energia Residual do Conjunto de Nós Atores para os Mecanismos de Coordenação Ator-Ator Baseados no Modelo de Propagação de Rádio *Free Space* em Redes com Diferentes Números de Atores

Mecanismo	12 atores	15 atores (s)	20 atores	30 atores
BPRC	$\sigma (J) = 59,9214$ $cv = 0,0755$	$\sigma (J) = 58,2316$ $cv = 0,0724$	$\sigma (J) = 52,5906$ $cv = 0,0661$	$\sigma (J) = 59,1370$ $cv = 0,0740$
BEPRC	$\sigma (J) = 60,8452$ $cv = 0,0767$	$\sigma (J) = 54,7255$ $cv = 0,0687$	$\sigma (J) = 55,1958$ $cv = 0,0682$	$\sigma (J) = 57,5620$ $cv = 0,0727$
BMLR	$\sigma (J) = 63,0462$ $cv = 0,0800$	$\sigma (J) = 60,0937$ $cv = 0,0760$	$\sigma (J) = 58,4831$ $cv = 0,0741$	$\sigma (J) = 60,3619$ $cv = 0,0762$
BEMLR	$\sigma (J) = 54,6220$ $cv = 0,0677$	$\sigma (J) = 50,6826$ $cv = 0,0628$	$\sigma (J) = 55,2740$ $cv = 0,0670$	$\sigma (J) = 58,8216$ $cv = 0,0731$

de atores, os quais são apresentados na Tabela 14. Portanto, tais resultados não permitem o traçado de conclusões a respeito da escalabilidade energética de ambos os mecanismos. No entanto, esses resultados sugerem que o BPRC permite o balanceamento uniforme do dispêndio energético entre os nós atores, o que pode acarretar no aumento do tempo de vida da rede. Tal comportamento não se reproduz de forma tão precisa no BEPRC, embora ambos os mecanismos apresentem um grau de variabilidade com certo nível de similaridade no que se refere ao perfil de consumo de energia. No que se refere ao BMLR e ao BEMLR, os resultados evidenciam que a energia residual média do conjunto de atores no BEMLR é maior do que no BMLR. Tais resultados sugerem que o ambos os mecanismos não apresentam escalabilidade energética. Os resultados experimentais demonstram ainda que a quantidade de energia residual do conjunto de nós atores no BMLR varia de maneira pouco significativa conforme ocorre o aumento do número de atores. Por conseguinte, sugere-se que esse mecanismo de coordenação consiga redistribuir o consumo de energia de forma uniforme entre os nós que compõem a rede, o que pode implicar no aumento do seu tempo de vida. Tal comportamento pode ser verificado nos valores assumidos pelas medidas de dispersão computadas para o BMLR, conforme a Tabela 13. De acordo com essa tabela, os coeficientes de variação do BMLR assumem valores que possuem um grau maior de homogeneidade, quando comparados com os coeficientes de variação calculados para o BEMLR.

Por outro lado, o impacto do modelo de propagação de rádio *Shadowing* e do número variável de atores sobre as medidas de desempenho dos mecanismos de coordenação ator-ator desenvolvidos é analisado a seguir, o qual é mensurado pelos resultados apresentados na Figura 23.

Segundo os resultados contidos na Figura 23(a), pode-se constatar que o tempo de resposta médio do BPRC é superior ao tempo de resposta médio do BEPRC, e que o tempo de resposta médio de ambos os mecanismos não aumenta de forma significativa de acordo com o aumento do número de atores. Portanto, não são escaláveis no que se refere a essa variável. Quanto aos resultados obtidos pelo BMLR e pelo BEMLR, verifica-se que o tempo de resposta médio do BEMLR é superior ao tempo de resposta médio do BMLR. Esses sugerem que o tempo de resposta médio do BMLR aumenta pelo aumento do número de atores, enquanto que o

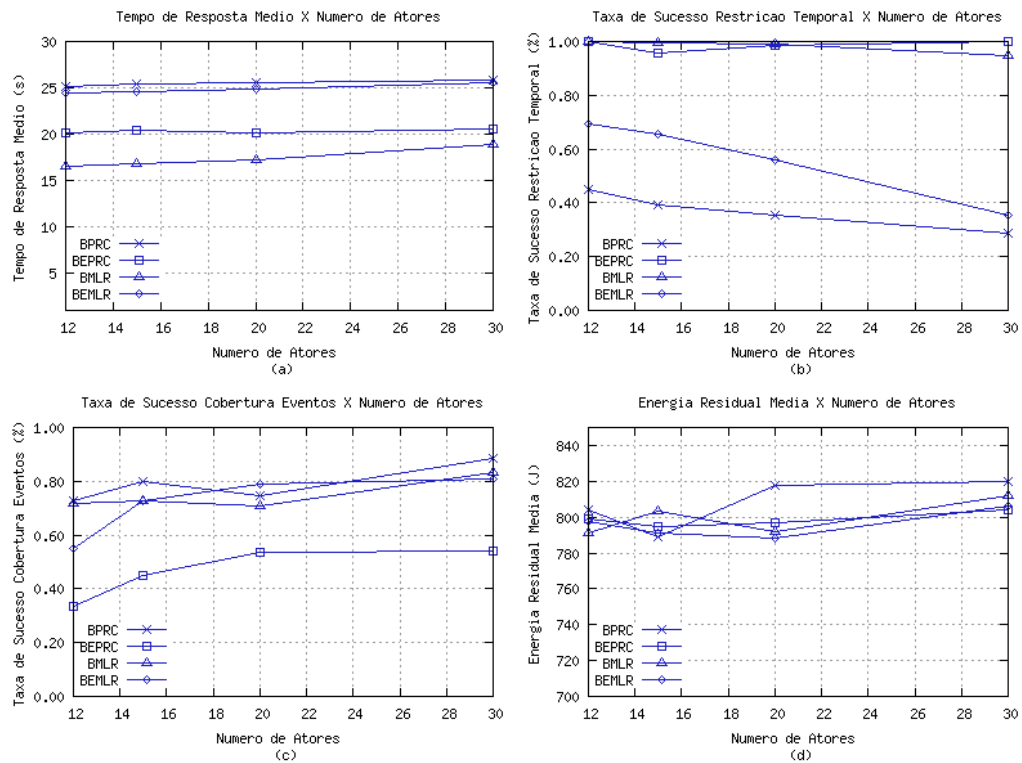


Figura 23 – Desempenho dos Mecanismos de Coordenação Ator-Ator para o Modelo de Propagação de Rádio *Shadowing* em Redes com Diferentes Números de Atores (a) Tempo de Resposta Médio (b) Taxa de Sucesso no Atendimento de Limitações Temporais (c) Taxa de Sucesso na Cobertura das Áreas de Eventos (d) Energia Residual Média

tempo de resposta médio do BEMLR não é afetado de forma tão significativa quanto o BMLR conforme aumenta-se o número de atores presentes nos experimentos de simulação. Logo, ambos os mecanismos não são escaláveis nos termos da medida de desempenho corrente. Tais resultados são semelhantes aos que foram obtidos a partir do emprego do modelo de propagação de rádio *Free Space*. Portanto, sugere-se que o uso de um modelo de propagação de rádio mais realístico como o *Shadowing* em conjunto com um número variável de atores estáticos não afeta o desempenho dos mecanismos propostos no que se refere ao tempo de resposta médio.

Tal tendência pode ser observada nos resultados apresentados na Figura 23(b). Esses evidenciam que a taxa de sucesso no atendimento de limitações temporais do BEPRC é maior do que a taxa de sucesso do BPRC, e que a referida taxa do BEPRC não aumenta de forma significativa com o aumento do número de atores. Todavia, esses resultados indicam que tal afirmação não é verdadeira para o BPRC, cuja a taxa diminui de forma expressiva quando aumenta-se o número de nós atores. Como implicação, ambos os mecanismos não manifestam um comportamento escalável no que se refere a essa taxa. Quanto ao BMLR e ao BEMLR, os resultados apontam que a taxa de sucesso no atendimento de limitações temporais do BMLR é maior do que a taxa de sucesso do BEMLR. Os resultados sugerem ainda que tal índice de desempenho não aumenta no BMLR conforme ocorre o aumento do número de nós atores, enquanto que tal índice de desempenho diminui de forma considerável no BEMLR quando aumenta-se o número de atores. Portanto, ambos não são escaláveis no que se refere a essa medida. Dessa forma, sugere-se que o emprego de um modelo de propagação de rádio mais realístico como o *Shadowing* em conjunto com um número variável de atores estáticos produz efeitos significativos sobre o desempenho dos mecanismos elaborados no que se refere à taxa de sucesso no atendimento de limitações temporais.

Conforme os resultados apresentados na Figura 23(c), verifica-se que a taxa de sucesso na cobertura das áreas de eventos do BPRC é maior do que a taxa de sucesso do BEPRC. Tais resultados sugerem que o desempenho do BPRC varia de maneira pouco considerável de acordo com o aumento do número de atores, enquanto que no BEPRC o desempenho aumenta quando ocorre o aumento do número desses. Embora tal tendência não possa ser verificada quando o número de atores varia de 20 para 30. Dessa forma, pode-se afirmar que os resultados experimentais não permitiram a identificação de uma tendência no comportamento desses mecanismos conforme aumenta-se o número de atores. Por conseguinte, tais resultados não permitem que se façam inferências a respeito da escalabilidade de tais mecanismos. Quanto aos resultados obtidos pelo BMLR e pelo BEMLR, constata-se que a taxa de sucesso na cobertura das áreas de eventos de cada mecanismo de colaboração varia de forma significativa e alternada conforme ocorre o aumento do número dos atores, sendo que os resultados dos experimentos de simulação não conseguiram demonstrar de forma efetiva uma tendência no comportamento desses. Provavelmente, não conseguiu-se atingir o comportamento em estado estacionário do sistema na simulação em particular desses mecanismos de coordenação, ou o uso de um modelo de propagação de rádio probabilístico tal como o *Shadowing* em conjunto com um número variável de

atores estáticos afetou o desempenho do BMLR e do BEMLR no que se refere à mencionada taxa.

Tabela 15 – Medidas de Variabilidade da Quantidade de Energia Residual do Conjunto de Nós Atores para os Mecanismos de Coordenação Ator-Ator Baseados no Modelo de Propagação de Rádio *Shadowing* em Redes com Diferentes Números de Atores

Mecanismo	12 atores	15 atores (s)	20 atores	30 atores
BPRC	$\sigma (J) = 57,5064$ $cv = 0,0715$	$\sigma (J) = 53,1120$ $cv = 0,0673$	$\sigma (J) = 49,9798$ $cv = 0,0611$	$\sigma (J) = 56,1332$ $cv = 0,0685$
BEPRC	$\sigma (J) = 54,8741$ $cv = 0,0687$	$\sigma (J) = 55,1781$ $cv = 0,0694$	$\sigma (J) = 55,2741$ $cv = 0,0694$	$\sigma (J) = 62,3878$ $cv = 0,0776$
BMLR	$\sigma (J) = 52,2729$ $cv = 0,0661$	$\sigma (J) = 59,5461$ $cv = 0,0741$	$\sigma (J) = 44,9587$ $cv = 0,0568$	$\sigma (J) = 59,9330$ $cv = 0,0738$
BEMLR	$\sigma (J) = 56,2095$ $cv = 0,0705$	$\sigma (J) = 51,2365$ $cv = 0,0648$	$\sigma (J) = 57,9072$ $cv = 0,0735$	$\sigma (J) = 51,7465$ $cv = 0,0642$

Segundo os resultados apresentados na Figura 23(d), constata-se que a energia residual média do conjunto de nós atores do BPRC é superior a energia residual média do BEPRC. Quanto ao efeito do número de atores sobre tais mecanismos, os resultados sugerem que o desempenho energético do BPRC aumenta de maneira levemente significativa de acordo com o aumento do número de atores, enquanto que no BEPRC tal desempenho se mantém levemente inalterado. Logo, tais mecanismos não manifestam a propriedade de escalabilidade energética. Tais resultados sugerem ainda que o BEPRC possibilita o balanceamento uniforme do consumo energético entre os nós atores, o que pode implicar no aumento do tempo de vida da rede. Tal comportamento não se reproduz no BPRC, o qual manifesta maior variabilidade no que se refere ao perfil de consumo de energia por parte dos nós atores. Tal tendência pode ser constatada a partir da comparação das medidas de dispersão (principalmente, do coeficiente de variação) de ambos os mecanismos para os diferentes números de atores, as quais são apresentadas na Tabela 15. De acordo com tais medidas, verifica-se que os valores computados para os coeficientes de variação do BEPRC apresentam menor variabilidade do que os valores computados para o BPRC quando o número de atores é igual a 12, 15 ou 20 (exceção ocorre quando número de atores é igual a 30), o que permite afirmar que o BEPRC manifesta um comportamento com um menor grau de variabilidade do que o BPRC no que se refere ao consumo dos recursos energéticos. Quanto ao desempenho obtido pelos mecanismos baseados na abordagem biológica, os resultados indicam que a energia residual média do conjunto de atores no BMLR é maior do que no BEMLR, com exceção do cenário composto por 12 atores, no qual a energia residual média do BEMLR é maior do que a energia residual média do BMLR. Tais resultados sugerem que ambos os mecanismos não apresentam escalabilidade energética. Complementarmente, esses resultados demonstram que no BEMLR a quantidade de energia residual dos nós atores varia de maneira pouco significativa conforme ocorre o aumento do número de atores. Portanto, sugere-se que esse mecanismo de colaboração nesse cenário de simulação particular consegue redistribuir o consumo de energia de forma uniforme entre os nós que compõem a rede, o que pode acarretar no aumento do seu tempo de vida. Tal comportamento pode ser verificado nos

valores assumidos pelas medidas de dispersão computadas para o BEMLR, conforme a Tabela 15. De acordo com essa tabela, verifica-se que os coeficientes de variação do BEMLR assumem valores que possuem um grau maior de homogeneidade entre si, quando comparados com os coeficientes de variação calculados para o BMLR. Esses variam de forma mais significativa, de acordo com o número de nós atores presentes nos experimentos. Logo, sugere-se que o emprego de um modelo de propagação de rádio mais realístico como o *Shadowing* em conjunto com um número variável de atores estáticos afeta o desempenho dos mecanismos elaborados no que se refere à energia residual média do conjunto de nós atores.



## 6 Conclusão

Em redes de sensores e atores sem fio surge o fenômeno da coordenação ator-ator. Esse consiste na seleção do ator ou do conjunto de atores que devem desempenhar uma tarefa no ambiente a ser controlado. Nesse processo, os atores devem se coordenar entre si da melhor forma possível para maximizar o desempenho global na execução da ação de controle. De acordo com essa definição, os nós atores devem se coordenar de forma autônoma para que decisões sejam tomadas e que ações sejam efetivadas dentro de restrições temporais pré-especificadas. A presente dissertação identificou estratégias e implementou mecanismos que visam à alocação dinâmica de tarefas no âmbito da referida coordenação.

O desempenho dos protocolos implementados foi avaliado por meio de simulações que tentaram reproduzir o funcionamento de uma rede de sensores e atores sem fio com certo nível de fidelidade. Os resultados sugerem que os protocolos desenvolvidos a partir do uso de estratégias econômicas e de estratégias biológicas possibilitam aos nós atores cooperarem entre si, o que pode viabilizar o desempenho de tarefas de forma autônoma para o controle automático das condições ambientais. Isso lhes permite reproduzir, em certo grau, aspectos de um comportamento emergente e adaptativo, o qual lhes possibilita lidar com as incertezas derivadas da interação com o ambiente.

Além disso, tais resultados sugerem ainda que as medidas de desempenho obtidas pelos algoritmos baseados em mecanismos de mercado possuem um certo grau de semelhança com as medidas de desempenho obtidas pelos algoritmos baseados no modelo de limiar de resposta. Cabe destacar que tal padrão de similaridade já havia sido identificado em publicações que propuseram mecanismos de coordenação para a alocação dinâmica de tarefas no contexto de sistemas multi-agentes [35] e no contexto de sistemas com múltiplos robôs móveis [74]. Inclusive, tal relação de equivalência entre esses modelos já havia sido discutida na literatura por meio de uma abordagem teórica. Essa argumenta que os algoritmos inspirados na divisão de trabalho em sociedades de insetos compartilham certas propriedades comuns com os algoritmos inspirados nos mecanismos de mercado, nos quais a alocação de tarefas é regulada por intermédio de um sistema de leilão [24]. Nesse modelo, o agente que submete o maior lance é o selecionado para a execução de tarefas. Um lance alto é equivalente a um limiar de resposta menor. Logo, pode-se sugerir que esses modelos possuem um certo grau de equivalência funcional, pois que ambos permitem a um dado sistema computacional complexo se adaptar a condições ambientais variáveis a partir do uso de informação local aos seus elementos constituintes.

## 6.1 Contribuições

A principal contribuição desta dissertação consiste na especificação e na implementação de algoritmos para a coordenação ator-ator por meio do emprego dos paradigmas econômico e biológico. Foram construídos dois mecanismos de coordenação baseados no reuso do *framework* de negociação incorporado no protocolo de redes de contratos (BPRC) e no reuso de uma de suas extensões (BEPRC). Além do mais, foi elaborado um mecanismo para a consecução de tal colaboração que se baseia em adaptações do modelo de limiar de resposta (BMLR) ao contexto de uma WSA. Adicionalmente, foi desenvolvido um mecanismo híbrido de coordenação que se baseia no reuso de um *framework* proposto para a alocação de tarefas em sistemas constituídos por múltiplos robôs que foi estendido para o contexto dos sistemas multi-agentes, o qual por sua vez foi adaptado ao âmbito de uma WSA (BEMLR).

Como contribuições secundárias, tem-se a definição de uma arquitetura genérica para a simulação de uma rede de sensores e atores sem fio. Ademais, essa arquitetura foi implementada por meio do desenvolvimento de extensões para o simulador de redes ns-2, as quais em seu conjunto denominam-se WSANSim. O código fonte do WSANSim bem como exemplos de *scripts* de simulação poderão ser disponibilizados na Internet, o que poderá permitir que outros pesquisadores utilizem-los em seus projetos de pesquisa.

## 6.2 Trabalhos Futuros

A seguir, discutem-se trabalhos futuros que poderão ser conduzidos em etapa posterior à finalização da presente dissertação.

### 6.2.1 Desenvolvimento de Extensões para o Protocolo de Redes de Contratos

De acordo com os resultados obtidos com a avaliação de desempenho do BPRC, sugere-se que o CNP não é adequado para o atendimento de restrições temporais. Entretanto, tal comportamento não se reproduz no BEPRC. O CNP surgiu no contexto da área dos sistemas multi-agentes, a qual é uma das principais ramificações de pesquisa da Inteligência Artificial Distribuída [111]. Contudo, sugere-se que o campo da Inteligência Artificial não tem se ocupado com questões de tempo real [70]. Portanto, poderá ser conduzida uma investigação *a posteriori* a respeito da introdução de extensões ao CNP em sua versão base que permitam a incorporação de limitações temporais ao seu princípio operativo. Ademais, sugere-se que o CNP funciona bem em ambientes que possuem um conjunto pequeno de agentes, porém isso não é verdadeiro quando o número de agentes aumenta [89]. Logo, argumenta-se que o CNP

apresenta problemas de escalabilidade. Segundo os resultados obtidos da avaliação do BPRC no que se refere à medida *taxa de sucesso no atendimento da restrição temporal*, tal afirmação encontra evidências que confirmam a sua validade. Contudo, esse comportamento parece não se evidenciar de forma tão perceptível quando se realiza a avaliação das medidas *taxa de sucesso na cobertura das áreas de eventos e quantidade de energia residual do conjunto de nós atores*. Portanto, experimentos adicionais poderão ser conduzidos para a verificação da generalidade de tal afirmação.

### 6.2.2 Coordenação Ator-Ator Tolerante a Falhas

O incremento no uso de computadores pela sociedade e o conseqüente incremento na confiança depositada nesses implica na necessidade de sistemas computacionais que possuam um alto grau de confiabilidade [87]. Existem muitas áreas nas quais os computadores desempenham atividades críticas. Dentre estas cabe mencionar os sistemas de controle de vôo, os sistemas de controle de tráfego aéreo e os sistemas de controle de processos industriais [16]. Uma rede de sensores e atores sem fio enquanto sistema distribuído embarcado estende tal cenário de aplicabilidade. Nesse contexto, a manifestação de defeitos em um componente do sistema pode conduzir a catástrofes das quais possam resultar perdas financeiras e talvez até mesmo a perda de vidas humanas [87]. Portanto, surge a necessidade de se projetar sistemas computacionais altamente confiáveis [167], o que pode ser alcançado por meio do emprego de técnicas de tolerância a falhas [18]. Visto que uma WSAAN pode ser aplicada em cenários de missão crítica, pode-se argumentar que o requisito de tolerância a falhas acrescenta-se aos requisitos de tempo real e de eficiência energética inerentes a tal sistema.

Um sistema é considerado tolerante a falhas se o comportamento desse é consistente com a sua especificação, apesar do aparecimento de defeitos em alguns dos seus componentes individuais [87]. Logo, se algum componente do sistema apresenta uma falha, então esse sistema é tolerante a falhas se o defeito do componente é mascarado, isto é, esse não é refletido no comportamento externo do sistema. A maioria das técnicas de tolerância a falhas procura mascarar a manifestação de defeitos no comportamento do sistema por meio do uso da técnica de redundância [87]. Essa consiste no emprego de um conjunto de componentes adicionais de *hardware* ou de *software* que não são necessários ao seu correto funcionamento. Entretanto, um componente redundante pode ser colocado em operação caso ocorra a manifestação de um defeito em algum componente primário do sistema.

A tolerância a falhas é composta por duas fases [18]: a detecção e a recuperação de erros. A primeira identifica a presença de um erro. Mais especificamente, deve existir algum mecanismo na fase de detecção de erros que permita ao sistema realizar a identificação de que um defeito se manifestou em algum dos seus componentes. Por sua vez, a segunda fase promove a transição de um estado que contém um ou mais erros para um estado no qual os erros não estejam mais

presentes no sistema, isto é, conduz-se o sistema a um estado consistente. A seguir descrevem-se algumas técnicas usualmente associadas a cada uma dessas fases no contexto dos sistemas distribuídos de propósito geral.

Na primeira fase, emprega-se um mecanismo de detecção de defeitos, o qual consiste na execução de um algoritmo distribuído que fornece evidências sobre a suspeita de defeitos a respeito de uma lista de processos que está sendo monitorada [87]. Em maior grau de detalhe, um detector de defeitos é um oráculo distribuído que fornece sugestões a respeito de quais processos possam estar manifestando um defeito: cada processo possui acesso a um módulo de detecção de defeitos que monitora outros processos do sistema, o qual mantém uma lista daqueles que ele correntemente suspeita de estarem manifestando um defeito [40] [41].

Na segunda fase, podem ser empregadas técnicas de *checkpointing/rollback*, protocolos de *commit* e técnicas de replicação de dados [87]. As técnicas de *checkpointing/rollback* visam à restauração do estado do sistema para um estado livre de erros. Essas se baseiam em pontos de recuperação, os quais são estabelecidos periodicamente durante a execução normal dos processos e que consistem no salvamento das informações necessárias para reinicializar um processo em uma área de armazenamento estável. As informações a serem salvas incluem os valores das variáveis do processo, o seu ambiente de execução, os valores dos registradores da máquina, etc. O armazenamento de tais informações na área de armazenamento estável garante que em caso de falha do processo, a informação salva está em um estado consistente. Quando um erro é detectado, o processo sofre um processo de *rollback*, isto é, ele é conduzido para um estado anterior consistente por meio da restauração das informações salvas no último *checkpoint* do referido processo. Por sua vez, os protocolos de *commit* são empregados para que se garanta que as ações executadas pelo sistema distribuído sejam atômicas, mesmo que ocorra a manifestação de defeitos em seus componentes. Nesse cenário, ou a ação é completada com sucesso por todos os processos do sistema, ou essa aparece como se nunca tivesse existido, de modo que o estado do sistema permanece inalterado. Dessa maneira, um protocolo de *commit* pretende assegurar que todos os nós que tomaram parte de uma transação distribuída realizem a confirmação da transação, ou o seu cancelamento. A finalidade de uma ação atômica é garantir que o conjunto de operações que a compõem sejam completadas com sucesso, ou em caso de falha, que essas apareçam para o sistema como se elas nunca tivessem sido executadas. Assim, não permite-se que ocorra a execução parcial do conjunto de operações caso constate-se a manifestação de defeitos no sistema. E finalmente, a abordagem empregada pelas técnicas de replicação de dados permite que uma ação seja completada com sucesso ainda que sejam constatadas falhas no comportamento do sistema. Analogamente ao que se verifica nos protocolos de *commit*, o objetivo de tais técnicas consiste na execução atômica das ações, com a diferença de que se está interessado na execução completa com sucesso das referidas ações, ao invés de se realizar o *rollback* do sistema, caso esse venha a manifestar defeitos.

Como consequência dessa visão de WSAAN enquanto sistema de missão crítica, poderá ser conduzido um levantamento a respeito de quais técnicas de tolerância a falhas podem ser apli-

cáveis ao contexto específico da coordenação ator-ator. Pois é nesse momento que a WSN efetivamente age sobre o ambiente monitorado, o qual é uma etapa crucial de operação do sistema. Particularmente, no que se refere aos mecanismos sugeridos para a referida coordenação no presente trabalho, pode-se constatar que os nós que desempenham o papel de gerente no BPRC e no BEPRC e de coletor no BEMLR representam um ponto único de falha no sistema, o que poderia ser resolvido pelo emprego da técnica de replicação de dados, por exemplo. Pois que mecanismos de tolerância a falhas pressupõem a existência de um mecanismo de detecção de defeitos [87], poderá ser conduzida uma investigação a respeito de que técnicas poderiam ser empregadas para se viabilizar a identificação dos nós atores ou dos dispositivos de atuação que venham a manifestar um defeito no decorrer da operação normal do sistema. Uma vez que os nós atores possuem maior capacidade computacional, de comunicação e energética do que os nós sensores, parece que é viável a elaboração de mecanismos de tolerância a falhas a partir da adaptação dos algoritmos acima discutidos ao cenário de uma WSN.

Visto que os nós de uma rede de sensores são propensos a manifestação de defeitos devido a exaustão da fonte de energia, a falhas de hardware e a erros no enlace de comunicação, sucede-se de que técnicas de tolerância a falhas focalizadas em tais redes têm sido introduzidas na literatura [94] [120]. Essas se dividem nas seguintes categorias principais: posicionamento dos nós, controle de topologia, detecção de eventos, coleta e agregação dos dados e gerenciamento de falhas. Contudo, esse cenário ainda não se reproduz em mesma escala no âmbito das redes de sensores e atores sem fio. Portanto, ainda existe espaço para contribuições nessa direção.

### 6.2.3 Mecanismos para a Coordenação Sensor-Sensor e Sensor-Ator

Na avaliação de desempenho dos protocolos propostos para a coordenação ator-ator, o protocolo LEACH foi utilizado tanto na coordenação sensor-sensor quanto na coordenação sensor-ator. No entanto, o LEACH apresenta limitações. Ele não pode fornecer garantias de que o número *default* de *cluster-heads* que se auto-elegem a cada rodada do protocolo, o qual é de 5%, consiga efetivamente cobrir toda a região de eventos que está sendo monitorada de maneira uniforme. Ademais, o LEACH não é focalizado em questões de tempo-real. Assim, poderá ser conduzida uma investigação a respeito da viabilidade de se implementar modificações no protocolo LEACH que o permitam garantir o atendimento de restrições temporais. Caso isso não seja viável, novos protocolos para a coordenação sensor-sensor e para a coordenação sensor-ator poderão ser elaborados, ou reusados da literatura, com o objetivo de se reavaliar o desempenho dos protocolos desenvolvidos para a coordenação ator-ator.

Nessa direção, verifica-se que o *framework Real-time Coordination and Routing (RCR)* [150] endereça o problema da coordenação sensor-ator e ainda possibilita a entrega de pacotes em tempo real tanto na arquitetura semi-automatizada quanto na arquitetura automatizada. Assim como o LEACH, ele é um protocolo de roteamento hierárquico baseado na clusterização

dinâmica dos nós sensores da rede em um conjunto de *clusters*, cujo objetivo consiste em se obter eficiência energética por meio da redistribuição de carga entre os nós sensores por meio da rotatividade do papel de *cluster-head*. A avaliação de desempenho do RCR foi conduzida por meio de experimentos no ns-2. No entanto, o código fonte resultante dessa implementação não está disponível publicamente, o que pode implicar na implementação a partir do zero do referido *framework*.

Complementarmente, em [43] é apresentado um protocolo de comunicação escalável, tolerante a falhas e que possui eficiência energética, o qual também é focado no contexto da coordenação sensor-ator. Nesse protocolo, os nós sensores reagem localmente após a ocorrência de modificações no contexto ambiental e interagem uns com os outros com o objetivo de ajustar o desempenho da rede em termos de energia, latência e taxa de sucesso na entrega dos pacotes que contém as informações sensoriadas. Por sua vez, os nós atores solicitam o encaminhamento de tais pacotes por meio da difusão de interesses, os quais são consultas que especificam os critérios que definem as informações que esses estão procurando obter na rede e que são disseminados dentro de uma área específica da rede de sensores. A avaliação de desempenho do presente protocolo também foi realizada por meio de experimentos no ns-2. Entretanto, como no caso do RCR, o código fonte resultante da simulação do referido protocolo não está acessível publicamente.

#### **6.2.4 Modelos de Mobilidade para Atores Móveis**

De acordo com recentes publicações, constata-se que a maioria dos trabalhos propostos na área de WSN emprega atores estáticos. Isso se deve ao fato de que não existe na literatura modelos específicos que possam reproduzir com certo grau de fidelidade a mobilidade desse tipo de agente artificial [96]. Na maioria dos projetos de pesquisa voltados para WSN é comum o emprego dos modelos de mobilidade *Random Walk*, *Random Waypoint* ou variações desses. No entanto, o movimento randômico de um nó ator pode não ser a melhor escolha para a maximização da detecção de eventos e subsequente ação no ambiente [96]. Logo, parece existir espaço para a investigação e para a validação de modelos de mobilidade aplicáveis a atores móveis.

#### **6.2.5 Migração do WSANSim para Outros Ambientes de Simulação**

Segundo os resultados relatados na subseção 4.5.2, sugere-se que o WSANSim consegue reproduzir com certo grau de fidelidade um ambiente baseado na tecnologia WSN. Porém, o WSANSim foi implementado no ns-2, o qual reconhecidamente não possui o grau de escalabilidade desejável para a simulação de redes compostas por um número relativamente grande de

nós [80]. Logo, as evidências parecem fornecer indícios de que o ns-2 não é a melhor opção para a simulação de redes WSN, e portanto, de redes WSAN. O pdns foi desenvolvido como o objetivo de superar a limitação do ns-2 no que refere ao requisito escalabilidade, assim como o J-Sim. Por isso, ambos são candidatos potenciais para a migração do WSANSim.

### **6.2.6 Implementação de Técnicas de Emulação no ns-2**

Verifica-se que existe uma tendência na literatura de se incorporar ambientes de rede reais aos ambientes de simulação por meio do uso de técnicas híbridas. Essa tendência pode ser nitidamente constatada tanto no SensorSim quanto no J-Sim. Tais técnicas visam ao aumento do grau de realismo dos experimentos conduzidos nos cenários simulados, o que pode se refletir no incremento da precisão dos resultados obtidos. Logo, poderão ser realizadas modificações no código do ns-2 que permitam o acréscimo de técnicas de emulação de rede. Cabe destacar que a versão corrente do ns-2 fornece uma facilidade de emulação simplificada que ainda está na fase experimental, a qual foi testada somente no sistema operacional FreeBSD versão 2.2.5.

### **6.2.7 Implementação de *Frontend* Gráfico para o WSANSim**

No estágio corrente, o usuário do ambiente WSANSim tem que escrever os seus próprios *scripts* na linguagem OTcl, a qual é a única maneira pela qual modelos de simulação podem ser interpretados pelo *kernel* do ns-2. No entanto, tal etapa do processo de simulação é sujeita a erros. Por conseguinte, poderá ser desenvolvida uma interface gráfica de usuário para a geração automática dos *scripts* de simulação.

## Referências

- [1] A. A. Abbasi, K. Akkaya, and M. Younis. A distributed connectivity restoration algorithm in wireless sensor and actor networks. In *Proceedings of the 32nd IEEE Conference on Local Computer Networks*, pages 496–503. IEEE Computer Society Press, 2007.
- [2] M. Abolhasan, T. Wysocki, and E. Dutkiewicz. A review of routing protocols for mobile ad hoc networks. *Ad Hoc Networks*, 2(1):1–22, 2004.
- [3] W. Agassounon and A. Martinoli. Efficiency and robustness of threshold-based distributed allocation algorithms in multi-agent systems. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1090–1097. ACM Press, 2002.
- [4] N. Ahmed, S. Kanhere, and S. Jha. Probabilistic coverage in wireless sensor networks. In *Proceedings of IEEE Conference on Local Computer Networks*, pages 672–681. IEEE Computer Society Press, 2005.
- [5] K. Akkaya, A. Thimmapuram, F. Senel, and S. Uludag. Distributed recovery of actor failures in wireless sensor and actor networks. In *Proceedings of the Wireless Communications and Networking Conference 2008*, pages 2480–2485. IEEE Computer Society Press, 2008.
- [6] S. Aknine. An extended multi-agent negotiation protocol. *Autonomous Agents and Multi-Agent Systems*, 8(1):5–45, 2004.
- [7] I. Akyildiz and I. Kasimoglu. Wireless sensor and actor networks: Research challenges. *Ad Hoc Networks Journal*, 2(4):351–367, 2004.
- [8] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.
- [9] J. N. Al-Karaki and A. E. Kamal. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*, 11(6):6–28, 2004.
- [10] E. Altman and T. Jiménez. Ns simulator for beginners. Capturado em: <http://www-sop.inria.fr/maestro/personnel/Eitan.Altman/ns.htm>, Abril 2008.
- [11] K. Arzen and A. Cervin. Control and embedded computing: Survey of research directions. In *Proceedings of 16th IFAC World Congress*, pages 342–353. Elsevier, 2005.
- [12] ASHRAE. *Thermal Environmental Conditions for Human Occupancy*. ASHRAE Learning Institute Publication, 30 p., 2004.
- [13] J. Aslam, Q. Li, and R. Rus. Three power-aware routing algorithms for sensor network. *Wireless Communications and Mobile Computing*, 3(2):187–208, 2003.



- [14] K. J. Astrom. Control system design. Capturado em: <http://www.cds.caltech.edu/murray/courses/cds101/fa02/caltech/astrom.html>, Setembro 2008.
- [15] K. J. Astrom and T. Haggund. *PID Controllers: Theory, Design, and Tuning*. Instrument Society of America, 2 edition, 343 p., 1995.
- [16] K. J. Astrom and B. Wittenmark. *Adaptive Control*. Addison-Wesley, 2 edition, 574 p., 1995.
- [17] K. J. Astrom and B. Wittenmark. *Computer-Controlled Systems: Theory and Design*. Prentice-Hall, 3 edition, 557 p., 1997.
- [18] A. Avizienis, J. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing*, 1(1):11–33, 2004.
- [19] P. Baldwin, S. Kohli, E. A. Lee, X. Liu, and Y. Zhao. Modeling of sensor nets in ptolemy ii. In *Proceedings of ACM/IEEE International Symposium on Information Processing in Sensor Networks*, pages 359–368. IEEE Computer Society Press, 2004.
- [20] M. Benchaiba, A. Bouabdallah, N. Badache, and M. Ahmed-Nacer. Distributed mutual exclusion algorithms in mobile ad hoc networks: an overview. *ACM SIGOPS Operating Systems Review*, 38(1):74–89, 2004.
- [21] G. Beni and J. Wang. Swarm intelligence. In *Proceedings of 7th Annual Meeting Robotics Society of Japan*, pages 425–428. RSJ Press, 1989.
- [22] R. H. Bishop. *The Mechatronics Handbook*. CRC Press, 1 edition, 1274 p., 2002.
- [23] R. H. Bishop. *Mechatronics an Introduction*. CRC Press, 320 p., 2006.
- [24] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: from Natural to Artificial Intelligence*. Oxford University Press, 307 p., 1999.
- [25] E. Bonabeau, A. Sobkowski, G. Theraulaz, and J. Deneubourg. *Bio-Computation and Emergent Computing*, chapter Adaptive Task Allocation Inspired by a Model of Division on Labor in Insect Societies, pages 36–45. World Scientific, 1997.
- [26] E. Bonabeau, G. Theraulaz, and J. Deneubourg. Quantitative study of the fixed threshold model for the regulation of division of labour in insect societies. *Proceedings of Royal Society of London*, 263(1376):1565–1569, 1996.
- [27] E. Bonabeau, G. Theraulaz, and J. Deneubourg. Fixed response thresholds and regulation of division of labour in insect societies. *Bulletin of Mathematical Biology*, 60:753–807, 1998.
- [28] G. Booch, J. Rumbaugh, and I. Jacobson. *The Unified Modeling Language User Guide*. Addison Wesley, 482 p., 1999.
- [29] S. C. Botelho and R. Alami. M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 2, pages 1234–1239. IEEE Computer Society Press, 1999.

- [30] L. A. Bryan and E. A. Bryan. *Programmable Controllers: Theory and Implementation*. American Technical, 2 edition, 1035 p., 2002.
- [31] N. Bulusu, J. Heidemann, and D. Estrin. Gps-less low cost outdoor localization for very small devices. *IEEE Personal Communications*, 7(5):28–34, 2000.
- [32] S. Burmester, H. Giese, and O. Oberschelp. *Informatics in Control, Automation and Robotics I*, chapter Hybrid UML Components for the Design of Complex Self-Optimizing Mechatronic Systems, pages 281–288. Springer, 2006.
- [33] A. Burns and A. Wellings. *Real-Time Systems and Programming Languages*. Addison-Wesley, 3 edition, 738 p., 2001.
- [34] N. W. Calderone and R. E. Page. Genotypic variability in age polyethism and task specialization in the honey bee *Apis mellifera*. *Behavioral Ecology and Sociobiology*, 22(1):17–25, 1988.
- [35] M. Campos, E. Bonabeau, G. Theraulaz, and J. L. Denenbourg. Dynamic scheduling and division of labor in social insects. *Adaptive Behavior*, 8(2):83–92, 2001.
- [36] Q. Cao, T. Yan, T. Abdelzaher, and J. Stankovic. Analysis of target detection performance for wireless sensor networks. In *Proceedings of International Conference on Distributed Computing in Sensor Networks*, pages 276–292. Springer, 2005.
- [37] T. W. Carley. Sidh: A wireless sensor network simulator. Technical Report TR 2005-88, Department of Electrical and Computer Engineering, University of Maryland, 2004.
- [38] A. Cavallo. *Using MATLAB, SIMULINK, and Control System Toolbox*. Prentice-Hall, 416 p., 1992.
- [39] T. Champ, J. Boleng, and V. Davis. A survey of mobility models for ad hoc network research. *Wireless Communications & Mobile Computing*, 2(5):483–502, 2002.
- [40] T. D. Chandra and S. Toueg. Unreliable failure detectors for reliable distributed systems. *Journal of ACM*, 43(2):235–267, 1996.
- [41] T. D. Chandra and S. Toueg. The weakest failure detector for solving consensus. *Journal of ACM*, 43(4):685–722, 1996.
- [42] J. Chang and L. Tassiulas. Routing for the maximum system lifetime in wireless ad-hoc networks. In *Proceedings of 37 th Annual Allerton Conference on Communications, Control and Computing*, pages 103–112. IEEE Computer Society Press, 1999.
- [43] I. Chatzigiannakis, A. Kinalis, and S. Nikolettseas. Priority based adaptive coordination of wireless sensors and actors. In *Proceedings of the 2nd ACM International Workshop on Quality of Service and Security for Wireless and Mobile Networks*, pages 37–44. ACM Press, 2006.
- [44] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *Wireless Communications*, 8(5):481–494, 2002.

- [45] G. Chen, J. Branch, M. Pflug, L. Zhu, and B. K. Szymanski. *Advances in Pervasive Computing and Networking*, chapter Sense: a Wireless Sensor Network Simulator, pages 249–267. Springer, 2006.
- [46] W. W. Chu, L. J. Holloway, M. Lan, and K. Efe. Task allocation in distributed data processing. *IEEE Computer*, 13(11):57–69, 1980.
- [47] V. A. Cicirello and S. F. Smith. Wasp-like agents for distributed factory coordination. *Autonomous Agents and Multi-Agent Systems*, 8(3):237–266, 2004.
- [48] L.P. Clare, E. H. Jennings, and J. L. Ciao. Performance evaluation modeling of networked sensors. In *Proceedings of IEEE Aerospace Conference*, pages 1313–1322. IEEE Computer Society Press, 2003.
- [49] National Semiconductor Corporation. Data sheet for lm135 series precision temperature sensors. Capturado em: <http://www.national.com/mpf/LM/LM135.html>, Outubro 2008.
- [50] Shih Hsing Industrial Corporation. Open frame solenoid sh-0415. Capturado em: <http://solenoid.products.com.tw/SH-0415.htm>, Agosto 2008.
- [51] G. F. Coulouris, J. Dollimore, and T. Kindberg. *Distributed Systems: Concepts and Design*. Addison-Wesley, 664 p., 1994.
- [52] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai. Ieee 802.11 wireless local area networks. *IEEE Communications Magazine*, 35(9):116–126, 1997.
- [53] J. Czyzyk, M. P. Mesnier, and J. J. Moré. The neos server. *IEEE Journal on Computational Science and Engineering*, 5(3):68–75, 1998.
- [54] R. Davis and R. G. Smith. Negotiation as a metaphor for distributed problem solving. *Distributed Artificial Intelligence*, 20(1):63–109, 1988.
- [55] H. M. Deitel and P. J. Deitel. *Java How to program*. Prentice-Hall, 1201 p., 2000.
- [56] UC Berkeley ECCS Departament. The ptolemy project. Capturado em: <http://ptolemy.eecs.berkeley.edu>, Agosto 2008.
- [57] C. Detrain and J. M. Pasteels. Caste differences in behavioral thresholds as basis for polyethism during food recruitment in the ant *Pheidole pallidula*. *Journal of Insect Behavior*, 4(2):157–176, 1991.
- [58] C. Detrain and J. M. Pasteels. Caste polyethism and collective defense in the ant *Pheidole pallidula*: the outcome of quantitative differences in recruitment. *Behavioral Ecology and Sociobiology*, 29(6):405–412, 1992.
- [59] M. B. Dias and A. Stenz. A market approach to multirobot coordination. Technical Report CMU-RI-TR-01-26, The Robotics Institute - Carnegie Mellon University, 2001.
- [60] R. C. Dorf and R. H. Bishop. *Modern Control Systems*. Prentice Hall, 659 p., 1998.
- [61] F. Dressler. Self-organization in ad hoc networks: Overview and classification. Technical Report Technical Report 02/06, University of Erlangen, Department of Computer Science, 2006.

- [62] W. Drozd. Automatically defined swarms for task allocation. In *Proceedings of IEEE International Conference on Intelligent Agent Technology*, pages 67–71. IEEE Computer Society Press, 2007.
- [63] J. Farines e J. S. Fraga e R. S. Oliveira. *Sistemas de Tempo Real*. Universidade Federal de Santa Catarina, 193 p., 2000.
- [64] K. Efe. Heuristic models of task assignment scheduling in distributed systems. *IEEE Computer*, 15(6):50–56, 1982.
- [65] H. El-Rewini, H. H. Ali, and T. Lewis. Task scheduling in multiprocessing systems. *IEEE Computer*, 28(12):27–37, 1995.
- [66] L. M. Feeney. A taxonomy for routing protocols in mobile ad hoc networks. Technical Report T99–07, Swedish Institute of Computer Science, 1999.
- [67] D. Ferguson, Y. Yemini, and C. Nikolaou. Microeconomic algorithms for load balancing in distributed computer systems. In *Proceedings of 8 th International Conference on Distributed Computing Systems*, pages 491–499. IEEE Computer Society Press, 1988.
- [68] R. Fourer, D. M. Gay, and B. W. Kernighan. *AMPL: A Modelling Language for Mathematical Programming*. Duxbury Press, 538 p., 2002.
- [69] R. A. Gagliano, M. D. Fraser, and M. E. Schaefer. Auction allocation for computing resources. *Communications of ACM*, 38(6):88–102, 1995.
- [70] A. Garvey and V. Lesser. A survey of research in deliberative real-time artificial intelligence. *Real-Time Systems*, 6(3):317–347, 1994.
- [71] B. P. Gerkey and M. J. Mataric. A market-based formulation of sensor-actuator network coordination. In *Proceedings of AAAI Spring Symposium in Intelligent Embedded and Distributed Systems*, pages 21–26. AAAI Press, 2002.
- [72] B. P. Gerkey and M. J. Mataric. Sold!: Auction methods for multirobot coordination. *IEEE Transactions on Robotics and Automation*, 18(5):758–768, 2002.
- [73] D. M. Gordon. The organization of work in social insect colonies. *Complexity*, 8(1):43–46, 2002.
- [74] J. Guerrero and G. Oliver. Multi-robot task allocation strategies using auction-like mechanisms. In *Proceedings of Sixth Congress of the Catalan Association for Artificial Intelligence*, pages 111–122. IOS Press, 2003.
- [75] V. C. Gungor. *Real-Time and Reliable Communication in Wireless Sensor and Actor Networks*. PhD thesis, School of Electrical and Computer Engineering, Georgia Institute of Technology, 2007.
- [76] M. Haenggi. Mobile sensor-actuator networks: Opportunities and challenges. In *Proceedings of the 2002 7th IEEE International Workshop on Cellular Neural Networks and Their Applications*, pages 283–290. IEEE Computer Society Press, 2002.
- [77] F. A. Hayek. The use of knowledge in society. *The American Economic Review*, 35(4):519–530, 1945.

- [78] M. Hefeeda and H. Ahmadi. Probabilistic coverage in wireless sensor networks. Technical Report TR 2006-21, School of Computer Science, Simon Fraser University, 2007.
- [79] W. R. Heinzelmann, A. Chandrakasan, and H. Balakrishnam. Energy efficient communication protocol for wireless microsensor networks. In *Proceedings of 33 rd Hawaii International Conference on System Sciences*, pages 3005–3014. IEEE Computer Society Press, 2000.
- [80] L. Hogue, P. Bouvry, and F. Guinand. An overview of manets simulation. *Electronic Notes in Theoretical Computer Science*, 150(1):81–101, 2006.
- [81] B. A. Huberman. Computation as economics. In *Second International Conference in Economics and Finance*. Social Science Electronic Publishing, 1996.
- [82] B. A. Huberman and T. Hogg. *The Ecology of Computation*, chapter The Behavior of Computational Ecologies, pages 77–115. North-Holland Press, 1988.
- [83] B. A. Huberman and T. Hogg. *SFI 1992 Lectures in Complex Systems*, chapter The Emergence of Computational Ecologies, pages 163–183. Addison-Wesley, 1993.
- [84] B. A. Huberman and T. Hogg. Distributed computation as an economic system. *Journal of Economic Perspectives*, 9(1):141–152, 1995.
- [85] M. N. Huhns and L. M. Stephens. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, chapter Multiagent Systems and Societies of Agents, pages 1–42. MIT Press, 1999.
- [86] R. Jain. *The Art of Computer Systems Performance Analysis*. John Wiley & Sons, 715 p., 1991.
- [87] P. Jalote. *Fault Tolerance in Distributed Systems*. Prentice Hall, 432 p., 1994.
- [88] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, volume 353, pages 153–181. Kluwer Academic Publishers, 1996.
- [89] Z. Juhasz and P. Paul. Scalability analysis of the contract net protocol. In *Proceedings of the 2nd International Symposium on Cluster Computing and the Grid*, pages 346–346. IEEE Computer Society Press, 2002.
- [90] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next century challenges: Mobile networking for smart dust. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 483–492. ACM Press, 1999.
- [91] K. Kar, M. Kodialam, T. V. Lakshman, and L. Tassiulas. Routing for network capacity maximization in energy-constrained ad-hoc networks. In *Proceedings of IEEE INFO-COMM*, pages 673–681. IEEE Computer Society Press, 2003.
- [92] L. J. Kazmier. *Schaum's Outline of Theory and Problems of Business Statistics*. McGraw-Hill, 375 p., 1982.
- [93] P. Klemperer. *Auctions: Theory and Practice*, chapter A Survey of Auction Theory, pages 9–57. Princeton University Press, 2004.

- [94] F. Koushanfar, M. Potkonjak, and A. Sangiovanni-Vincentelli. *Handbook of Sensor Networks*, chapter Fault Tolerance in Wireless Sensor Networks, pages 1–22. CRC Press, 2004.
- [95] S. Kraus. Automated negotiation and decision making in multiagent environments. *Multi-agents Systems and Applications*, 2086(2):150–172, 2001.
- [96] S. K. Krishnakumar and R. T. Abler. Intelligent actor mobility in wireless sensor and actor networks. In *Proceedings of 1st IFIP International Conference in Wireless Sensor and Actor Networks*, volume 248, pages 13–22. Springer, 2007.
- [97] T. R. Kurfess. *Robotics and Automation Handbook*. CRC Press, 618 p., 2005.
- [98] T. H. Labella, I. Dietrich, and F. Dressler. Baraka: A hybrid simulator of sanets. In *Proceedings of 2nd International Conference on Communication Systems Software and Middleware*, pages 1–8. IEEE Computer Society Press, 2007.
- [99] T. H. Labella and F. Dressler. A bio-inspired architecture for division of labour in sanets. In *Proceedings of the 1st International Conference on Bio Inspired Models of Network, Information and Computing Systems*, pages 1–8. IEEE Computer Society Press, 2006.
- [100] A. M. Law and W. D. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill, 760 p., 2000.
- [101] C. Lin, C. C. Federspiel, and D. M. Auslander. Multi-sensor single-actuator control of hvac systems. In *Proceedings of the International Conference of Embedded Building Operations*, pages 45–52. IEEE Computer Society Press, 2002.
- [102] B. Liu and D. Towsley. A study on the coverage of large-scale sensor networks. In *Proceedings of IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pages 475–483. IEEE Computer Society Press, 2004.
- [103] N. Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers, 872 p., 1996.
- [104] R. MacDowall. *Fundamentals of HVAC Systems*. Elsevier, 235 p., 2006.
- [105] P. Maille and B. Tuffin. Multi-bid auctions for bandwidth allocation for communication networks. In *Proceedings of IEEE INFOCOM*, pages 52–65. IEEE Computer Society Press, 2004.
- [106] M. J. Mataric, G. S. Sukhatme, and E. H. Ostergaard. Multi-robot task allocation in uncertain environments. *Autonomous Robots*, 14(2):255–263, 2003.
- [107] J. G. McNeff. The global positioning system. *IEEE Transactions on Microwave Theory and Techniques*, 50(3):645–652, 2002.
- [108] T. Melodia, D. Pompili, V. C. Gungor, and I. F. Akyildiz. Communication and coordination in wireless sensor and actor networks. *IEEE Transactions on Mobile Computing*, 6(10):1116–1129, 2007.
- [109] M. S. Miller and K. E. Drexler. *The Ecology of Computation*, chapter Market and Computation: Agoric Open Systems, pages 133–167. North-Holland Press, 1988.

- [110] C. Mitidieri, C. E. Pereira, and L. L. Penz. Real time issues on coordinating cooperative autonomous objects. In *Proceedings of 7th International Workshop on Object-Oriented Real-Time Dependable Systems*, pages 131–138. IEEE Computer Society Press, 2002.
- [111] B. Moulin and B. Chaib-Draa. *Foundations of Distributed Artificial Intelligence*, chapter An Overview of Distributed Artificial Intelligence, pages 3–55. John Wiley and Sons Inc., 1996.
- [112] J. Muller. *Foundations of Distributed Artificial Intelligence*, chapter Negotiation Principles, pages 211–229. John Wiley and Sons Inc., 1996.
- [113] R. M. Murray, editor. *Control in an Information Rich World: Report of the Panel on Future Directions in Control, Dynamics, and Systems*. Society for Industrial and Applied Mathematics, 104 p., 2003.
- [114] R. M. Murray, K. J. Astrom, S. P. Boyd, R. W. Brockett, and G. Stein. Control in an information rich world. *IEEE Control Systems Magazine*, 23(2):20–33, 2003.
- [115] Networked and Embedded Systems Laboratory. Sensorsim: A simulation framework for sensor networks. Capturado em: <http://nesl.ee.ucla.edu/projects/sensorsim>, Outubro 2008.
- [116] E. C. H. Ngai, M. R. Lyu, and J. Liu. A real-time communication framework for wireless sensor-actuator networks. In *Proceedings of IEEE Aerospace Conference 2006*, pages 1–9. IEEE Computer Society Press, 2006.
- [117] The Network Simulator NS2. Capturado em: <http://www.isi.edu/nsnam/ns/>, Abril 2008.
- [118] U.S. Department of Energy. *Instrumentation and Control Fundamentals Handbook*, volume 2. 168 p., 1992.
- [119] K. Ogata. *Modern Control Engineering*. Prentice-Hall, 997 p., 2001.
- [120] L. Paradis and Q. Han. A survey of fault management in wireless sensor networks. *Journal of Network and Systems Management*, 15(2):171–190, 2007.
- [121] J. Park and S. Sahni. An online heuristic for maximum lifetime routing in wireless sensor networks. *IEEE Transactions on Computers*, 55(8):1048–1056, 2006.
- [122] S. Park, A. Savvides, and M. B. Srivastava. Sensorsim: A simulation framework for sensor networks. In *Proceedings of the 3rd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 104–111. ACM Press, 2000.
- [123] S. Park, A. Savvides, and M. B. Srivastava. Simulating networks of wireless sensors. In *Proceedings of 2001 Winter Simulation Conference*, pages 1330–1338. ACM Press, 2001.
- [124] D. W. Pearce, editor. *The MIT Dictionary of Modern Economics*. MIT Press, 486 p., 1986.
- [125] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In *Proceedings of 2nd Workshop on Mobile Computing Systems and Applications*, pages 90–100. ACM Press, 1999.

- [126] R. S. Pindyck and D. L. Rubinfeld. *Microeconomics*. Macmillan Publishing Company, 791 p., 1991.
- [127] The Rice University Monarch Project. Capturado em: <http://www.monarch.cs.rice.edu/>, Maio 2008.
- [128] The VINT Project. The ns manual. Capturado em: <http://www.isi.edu/nsnam/ns/doc/index.html>, Outubro 2008.
- [129] MODBUS Industrial Network Protocol. Capturado em: <http://www.modbus.org>, Novembro 2008.
- [130] PROFIBUS Industrial Network Protocol. Capturado em: <http://www.profibus.com>, Novembro 2008.
- [131] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava. Energy-aware wireless microsensor networks. *IEEE Signal Processing*, 19(2):40–50, 2002.
- [132] B. Randell, J. C. Laprie, H. Kopetz, and B. Littlewood, editors. *Predictably Dependable Computing Systems*. Springer, 608 p., 1995.
- [133] T. S. Rappaport. *Wireless communications, principles and practice*. Prentice Hall, 641 p., 1996.
- [134] G. F. Riley, R. M. Fujimoto, and M. H. Ammar. A generic framework for parallelization of network simulations. In *Proceedings of the 7th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pages 128–135. IEEE Computer Society Press, 1999.
- [135] G. E. Robinson. Modulation of alarm pheromone perception in the honey bee: Evidence for division of labor based on hormonally regulated response thresholds. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 160(5):613–619, 1987.
- [136] G. E. Robinson. Regulation of division of labor in insect societies. *Annual Review of Entomology*, 37(1):637–665, 1992.
- [137] G. E. Robinson, R. E. Page, and Z. Y. Huang. Temporal polyethism in social insects is a developmental process. *Animal Behavior*, 48(2):467–469, 1994.
- [138] Robertino Robots. Capturado em: <http://www.openrobertino.org>, Outubro 2008.
- [139] E. Royer and C. Toh. A review of current routing protocols for ad-hoc mobile wireless networks. *IEEE Personal Communications*, 6(2):46–55, 1999.
- [140] L. B. Ruiz, J. M. S. Nogueira, and A. A. F. Loureiro. *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, volume 1, chapter Sensor Network Management. CRC Press, 2004.
- [141] L. B. Ruiz, I. G. Siqueira, L. B. e Oliveira, H. C. Wong, J. M. S. Nogueira, and A. A. F. Loureiro. Fault management in event-driven wireless sensor networks. In *Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 149–156. ACM Press, 2004.



- [142] D. Rutherford, editor. *Dictionary of Economics*. Routledge, 1393 p., 1992.
- [143] T. W. Sandholm. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, chapter Distributed Rational Decision Making, pages 201–258. MIT Press, 1999.
- [144] C. Schindelhauer. Mobility in wireless networks. In *Proceedings of 32nd International Conference on Current Trends in Theory and Practice of Computer Science*, pages 21–27. Springer, 2006.
- [145] P. S. Schneider and D. H. Eberly. *Geometric Tools for Computer Graphics*. Morgan Kaufmann, 1007 p., 2003.
- [146] T. D. Seeley. Social foraging in honey bees: How nectar foragers assess their colony’s nutritional status. *Behavioral Ecology and Sociobiology*, 24(3):181–199, 1989.
- [147] T. D. Seeley. The tremble dance of the honey bee: Message and meanings. *Behavioral Ecology and Sociobiology*, 31(6):375–383, 1992.
- [148] F. Senel, K. Akkaya, and M. Younis. An efficient mechanism for establishing connectivity in wireless sensor and actor networks. In *IEEE Global Telecommunications Conference 2007*, pages 1129–1133. IEEE Computer Society Press, 2007.
- [149] World Weather Information Service. Capturado em: <http://www.worldweather.org>, Setembro 2008.
- [150] G. A. Shah, M. Bozyigit, O. B. Akan, and B. Baykal. Real-time coordination and routing in wireless sensor and actor networks. In *Proceedings of the 6th International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking*, pages 365–383. Springer, 2006.
- [151] Mannasim Framework Wireless Sensor Networks Simulator. Capturado em: <http://www.mannasim.dcc.ufmg.br/>, Março 2008.
- [152] P. K. Sinha. *Distributed Operating Systems*. IEEE Computer Society Press, 743 p., 1997.
- [153] R. Smith. Open dynamics engine. Capturado em: <http://ode.org>, Setembro 2008.
- [154] R. G. Smith. A framework for problem solving in a distributed processing environment. Technical report, Department of Computer Science, Stanford University, 1978.
- [155] R. G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computer*, 29(12):1104–1113, 1980.
- [156] R. G. Smith and R. Davis. Frameworks for cooperation in distributed problem solving. *IEEE Transactions on Systems, Man and Cybernetics*, 11(1):61–70, 1981.
- [157] S. W. Smith. The scientist and engineer’s guide to digital signal processing. Capturado em: <http://www.dspguide.com/editions.htm>, Maio 2008.
- [158] A. Sobeih, W. Chen, J. C. Hou, L. Kung, N. Li, H. Lim, H. Tyan, and H. Zhang. J-sim: a simulation and emulation environment for wireless sensor networks. *IEEE Wireless Communications*, 13(4):104–119, 2006.

- [159] A. Sobeih and J. C. Hou. A simulation framework for sensor networks in j-sim. Technical Report Technical Report UIUCDCS-2003-2386, Department of Computer Science, University of Illinois at Urbana-Champaign, 2003.
- [160] S. Srinivasan and N. K. Jha. Safety and reliability driven task allocation in distributed systems. *IEEE Transactions on Parallel and Distributed Systems*, 10(3):238–251, 1999.
- [161] J. A. Stankovic. Misconceptions about real-time computing. *IEEE Computer*, 21(10):10–19, 1988.
- [162] M. Stemm and R. H. Katz. Measuring and reducing energy consumption of network interfaces in hand-held devices. *IEEE Transactions on Communications*, E80-B(8):1125–1131, 1997.
- [163] I. Stojmenovic. Data communication and coordination in wireless sensor and sensor-actuator networks. In *International Conference on Sensor Technologies and Applications*, pages 122–125. IEEE Computer Society Press, 2007.
- [164] B. Sundararaman, U. Buy, and A. D. Kshemkalyani. Clock synchronization for wireless sensor networks: a survey. *Ad Hoc Networks*, 3(3):281–323, 2005.
- [165] S. Sundresh, W. Kim, and G. Agha. Sens: A sensor, environment and network simulator. In *Proceedings of 37th Annual Simulation Symposium*, pages 221–228. IEEE Computer Society Press, 2004.
- [166] C. Szyperski. *Component Software: Beyond Object-Oriented Programming*. Addison-Wesley, 589 p., 1999.
- [167] G. Tel. *Introduction to Distributed Algorithms*. Cambridge University Press, 534 p., 1994.
- [168] G. Theraulaz, E. Bonabeau, and J. Deneubourg. Response threshold reinforcement and division of labour in insect societies. *Biological Sciences*, 265(1393):327–332, 1998.
- [169] G. Theraulaz, E. Bonabeau, and J. Deneubourg. Threshold reinforcement and the regulation of division of labour in insect societies. *Biological Sciences*, 266(1):327–335, 1998.
- [170] G. Theraulaz, S. Goss, J. Gervet, and J. L. Deneubourg. Task differentiation in polistes wasp colonies: a model for self-organizing groups of robots. In *Proceedings of First International Conference on Simulation of Adaptive Behavior*, pages 346–355. ACM Press, 1991.
- [171] OTcl Tutorial. Capturado em: <http://bmrc.berkeley.edu/research/cmt/cmtdoc/otcl/tutorial.html>, Março 2008.
- [172] A. Varga. The omnet++ discrete event simulation system. In *Proceedings of the 15h European Simulation Multiconference*, pages 319–324. SCS - European Publishing House, 2001.
- [173] H. R. Varian. *Intermediate Microeconomics: A Modern Approach*. W. W. Norton and Company, 778 p., 2003.

- [174] R. Vedantham. *Energy-Efficient Networks Protocols for Wireless Sensor and Actor Networks*. PhD thesis, School of Electrical and Computing Engineering, Georgia Institute of Technology, 2006.
- [175] R. Vedantham, Z. Zhuang, and R. Sivakumar. Mutual exclusion in wireless sensor and actor networks. In *3rd IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, pages 346–355. IEEE Computer Society Press, 2006.
- [176] C. A. Waldspurger, T. Hogg, B. A. Huberman, and J. O. Kephart. Spawn: A distributed computational economy. *IEEE Transactions on Software Engineering*, 18(2):103–117, 1992.
- [177] W. E. Walsh and M. P. Wellman. A market protocol for decentralized task allocation. In *Proceedings of 3th International Conference on Multi Agent Systems*, pages 325–332. IEEE Computer Society Press, 1998.
- [178] M. P. Wellmann and J. Q. Cheng. The walras algorithm: a convergent distributed implementation of general equilibrium outcomes. *Computational Economics*, 12(1):1–24, 1998.
- [179] E. O. Wilson. The sociogenesis of insect colonies. *Science*, 228(4707):1489–1495, 1985.
- [180] J. Wu, S. Yang, and M. Cardei. On maintaining sensor-actor connectivity in wireless sensor and actor networks. In *Proceedings of the 27th Conference on Computer Communications*, pages 286–290. IEEE Computer Society Press, 2008.
- [181] F. Xia, Y. Tian, Y. Li, and Y. Sun. Wireless sensor/actuator network design for mobile control applications. Capturado em: <http://www.mdpi.org/sensors/papers/s7102157.pdf>, Junho 2007.
- [182] S. Young. *Real Time Languages: Design and Development*. Ellis Horwood, 352 p., 1982.
- [183] H. Yuan, H. Ma, and H. Liao. Coordination mechanism in wireless sensor and actor networks. In *Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences*, pages 627–634. IEEE Computer Society Press, 2006.
- [184] Y. Zou and K. Chakrabarty. Sensor deployment and target localization in distributed sensor networks. *ACM Transactions on Embedded Computing Systems*, 3(1):61–91, 2004.
- [185] Y. Zou and K. Chakrabarty. A distributed coverage-and-connectivity-centric technique for selecting active nodes in wireless sensor networks. *IEEE Transactions on Computers*, 54(8):978–991, 2005.