

THAIS REGINA DE MOURA BRAGA

**UM ELEMENTO AUTONÔMICO PARA
REDES DE SENSORES SEM FIO**

Belo Horizonte
31 de agosto de 2006

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

**UM ELEMENTO AUTONÔMICO PARA
REDES DE SENSORES SEM FIO**

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

THAIS REGINA DE MOURA BRAGA

Belo Horizonte
31 de agosto de 2006



UNIVERSIDADE FEDERAL DE MINAS GERAIS

FOLHA DE APROVAÇÃO

Um Elemento Autônomo para
Redes de Sensores Sem Fio

THAIS REGINA DE MOURA BRAGA

Dissertação defendida e aprovada pela banca examinadora constituída por:

Ph. D. LINNYER BEATRYS RUIZ – Orientadora
Universidade Federal de Minas Gerais

Ph. D. ANTONIO ALFREDO FERREIRA LOUREIRO – Co-orientador
Universidade Federal de Minas Gerais

Ph. D. JOSÉ MARCOS SILVA NOGUEIRA
Universidade Federal de Minas Gerais

Ph. D. GERALDO ROBSON MATEUS
Universidade Federal de Minas Gerais

Belo Horizonte, 31 de agosto de 2006

Resumo

Redes Autônomicas controlam e supervisionam a si próprias sem intervenção humana direta. Na definição e desenvolvimento destas redes, a menor parte a ser construída é o Elemento Autônomico (EA). Os aspectos de projeto e as funcionalidades envolvidas na modelagem de um EA influenciam no desempenho dos elementos gerenciados, da rede e dos serviços provisionados. Este trabalho apresenta o modelo de um Elemento Sensor Autônomico (ESAs) para Redes de Sensores Sem Fio (RSSF). O desafio neste caso é projetar ESAs que possam ser executados em ambientes com restrições severas de hardware, software, comunicação de dados e energia, e que ainda assim possam promover a produtividade destes recursos e a qualidade dos serviços provisionados pelos nós sensores. Em particular, o trabalho apresenta a instância de um modelo de ESA considerando a plataforma de nós sensores Mica Motes 2. Os resultados mostram a relevância de se aplicar computação autônomicas em RSSFs. As simulações e experimentos realizados neste trabalho revelam que as redes cujos componentes possuem um ESA embutido são capazes de reduzir o consumo médio de energia, diminuir a porcentagem de pacotes perdidos e ainda melhorar parâmetros de qualidade de serviço, tais como produtividade e precisão na entrega das informações.

Abstract

Autonomic Networks control and supervise themselves without direct human intervention. In the definition and development of these networks, the smallest part to be built is the Autonomic Element (AE). The design aspects and the functionalities involved with the modelling of an AE influence the performance of the managed element, network and provided services. This work presents the model of an Autonomic Sensor Element (ASE) for Wireless Sensor Networks (WSNs). The challenge in this case is to design ASEs that can be executed in environments with severe hardware, software, data communication and energy restrictions, and that still can promote the productivity of these resources and the quality of the services provided by the sensor nodes. Particularity, this work creates an instance of an ASE model considering the Mica Motes 2 sensor nodes platform. The results have shown the relevance of applying autonomic computing in WSNs. The simulations and experiments performed in this work revealed that the networks with components that have an embedded ASE were able to decrease average energy consumption and packet loss percentage, and also improve quality of services parameters, like productivity and precision in information delivery.

Dedico esse trabalho aos meus amores: Fabrício e minha família.

Agradecimentos

Agradeço em primeiro lugar a Deus que, além de me dar a oportunidade de mais uma vida, permitiu que eu tivesse saúde, paz, felicidade e que compartilhasse meus dias com pessoas muito especiais: Fabrício, mamãe, papai, Matheus e Marcos.

Ao Fabrício, não tenho palavras para agradecer o amor, o companheirismo, a paciência, o carinho, o amparo e a dedicação. Ao longo desses anos você foi amigo, namorado e professor. Me ajudou nos momentos mais difíceis e compartilhou intensamente comigo as experiências mais felizes da minha vida. Compreendeu meus problemas e me apontou soluções. Foi meu ombro amigo onde encontrei espaço para dividir dúvidas, soluções, tristezas, alegrias, incertezas e certezas. Um milhão de vezes obrigada por tudo! Aproveito para agradecer a sua família: Geralda, Antônio, Cristiane, Flávio, Máisa e Luísa, por sempre me receberem com grande carinho e por apoiarem nossas escolhas ao longo desses anos.

A minha mamãe Ana Maria de Moura Braga, agradeço pelo amor incondicional, pelas orações, pelos abraços, beijos e carinhos, enfim, pelo papel de mãe exemplarmente cumprido. Ao meu papai Geraldo Clebs Braga agradeço também pelo amor e orações, mas principalmente pelos exemplos e pelo enorme alto-astrol, bom humor e disposição (principalmente pela manhã). Mãe e pai, vocês são muito importantes na minha vida. São meu porto-seguro, meus heróis, exemplos de vida. Obrigada por apoiarem todas as minhas decisões e tenham certeza de que só cheguei até aqui porque vocês foram, e continuam sendo, pais maravilhosos!

Aos meus irmãos Matheus e Marcos agradeço pelo companheirismo e pela convivência harmoniosa de sempre. Obrigada pelo apoio e principalmente pelos momentos de descontração. Graça a Deus tive a sorte de contar com irmãos que são meus amigos e que estão sempre ao meu lado me dando suporte necessário.

A minha orientadora Bia, agradeço pela acolhida no momento em que nada sabíamos, pela dedicação no processo de aprendizado e pela confiança quando finalmente pudemos ajudar e trabalhar juntos. Tenha certeza de que você foi muito importante em minha formação e que hoje sou uma pesquisadora graças a sua paciência e empenho. Ao meu co-orientador, professor Antonio Alfredo Ferreira Loureiro, agradeço inicialmente pela oportunidade dada para meu ingresso na carreira científica e também pelos preciosos conhecimentos e experiências compartilhadas. Agradeço ainda aos professores José Marcos Silva Nogueira e Geraldo Robson Matheus pelos importantes comentários e sugestões dadas para a melhoria desse trabalho.

Agradeço ainda pela amizade e pelo conhecimento compartilhado com amigos e amigas do Manna, SIS, ATM, SensorNet, TBb. Em particular agradeço a Ana Paula e Isabela pelas

consultorias, pelas conversas produtivas e improdutivas, pelo apoio nas campanhas de Natal, enfim pela grande amizade cultivada nos anos de convivência lado a lado! Agradeço a Mayara pela ajuda no desenvolvimento dessa dissertação. Enfim, agradeço a todos que contribuíram de alguma forma na elaboração desse trabalho e em minha formação profissional.

Sumário

1	Introdução	1
1.1	Redes de Sensores Sem Fio	1
1.2	Nós Sensores Sem Fio	3
1.2.1	Plataforma de Nós Sensores Sem Fio Mica Motes 2	5
1.3	Computação Autônômica	7
1.4	Motivações	9
1.5	Objetivos	10
1.6	Contribuições	11
1.7	Organização	11
1.8	Considerações Finais	12
2	Redes Autônômicas	13
2.1	Definição e Motivações	13
2.2	Redes Autônômicas Sem Fio	14
2.3	Níveis de Autonomia	16
2.4	Serviços de Auto-gerenciamento	18
2.5	Elemento Autônômico (EA)	21
2.5.1	Elementos Gerenciados	22
2.5.2	Gerente Autônômico	22
2.5.3	Serviços de um Gerente Autônômico	24
2.5.4	Ciclo de Vida de um EA	31
2.5.5	Ciclo de Relacionamento entre EAs	32
2.6	Construção de uma Rede Autônômica	32
2.7	Considerações Finais	35
3	Trabalhos Relacionados	36
3.1	Computação Autônômica para Sistemas Tradicionais	36
3.2	Computação Autônômica Aplicada a Redes Com Fio	38
3.3	Computação Autônômica Aplicada a Redes Sem Fio e RSSFs	40
3.4	Arquitetura Manna	43
3.5	Gerenciamento Tradicional	45
3.5.1	Arquiteturas de Gerenciamento Tradicional	47

3.6	Considerações Finais	49
4	Elemento Sensor Autônomo	51
4.1	Visão Geral do Gerente Autônomo de um ESA	51
4.2	Serviço de Monitoração	52
4.3	Serviço de Análise	54
4.4	Serviço de Planejamento	57
4.5	Serviço de Execução	58
4.6	Base de Conhecimento (BC)	60
4.7	Outros Aspectos	62
4.8	Considerações Finais	63
5	Projeto de Simulação	65
5.1	Metodologia	65
5.2	Instância do ESA para Plataforma Mica Motes 2	67
5.2.1	Suposições Consideradas	69
5.2.2	Requisitos de Memória	70
5.2.3	Ferramenta de Simulação	71
5.2.4	Classificação de Dados	71
5.2.5	Correlação de Eventos	72
5.2.6	Políticas para Reprogramação e Reconfiguração	74
5.2.7	Mensagens Utilizadas	75
5.3	Descrição dos Parâmetros de Simulação	76
5.4	Considerações Finais	78
6	Análise de Resultados	80
6.1	Consumo de Energia	80
6.2	Informações Enviadas pelos Nós Comuns	81
6.3	Informações Recebidas pelo PA	83
6.4	Número de Pacotes Perdidos	84
6.5	Consumo de Memória	85
6.6	Variação do Intervalo de Monitoração	86
6.7	Variação do Número de Eventos	89
6.8	Variação da Quantidade de Processamento para Execução do Ciclo Autônomo	93
6.9	Considerações Finais	95
7	Conclusão	98
A	Projeto Experimental para Análise Funcional	101
A.1	Cenários Testados e Resultados Obtidos	104
A.2	Considerações Finais	106
	Referências Bibliográficas	109

Lista de Figuras

1.1	Rede de Sensores Sem Fio (RSSF).	2
1.2	Apresentação de alguns cenários de aplicações para RSSFs.	3
1.3	Principais componentes do nó sensor sem fio [Ruiz, 2003].	4
1.4	Ilustração de nós sensores da família Mica Motes.	6
1.5	Atuação do sistema autonômico nervoso em seres humanos.	8
2.1	Classificação indicativa do nível de autenticidade de uma rede [Ganek e Corbi, 2003].	17
2.2	Principais serviços autonômicos: auto-configuração, auto-otimização, auto-proteção, auto-cura.	19
2.3	Estrutura Interna de um elemento gerenciado [AC Blueprint, 2006].	23
2.4	Estrutura Interna de um Elemento Autonômico [Kephart e Chess, 2003].	23
2.5	Arcabouço PMAC comparado a arquitetura IETF/DMTF tradicional.	27
3.1	Gerenciamento tridimensional proposto pela arquitetura Manna.	44
3.2	Relacionamento entre serviços, funções e modelos de gerenciamento.	44
4.1	Visão geral dos serviços, funções e componentes de conhecimento que compõem o Elemento Sensor Autonômico.	52
4.2	Funções que compõem o serviço de monitoração de um ESA.	53
4.3	Funções que compõem o serviço de análise de um ESA.	55
4.4	Funções que compõem o serviço de planejamento de um ESA.	57
4.5	Funções que compõem o serviço de execução de um ESA.	59
5.1	Instância do modelo de ESA implementada para as simulações deste trabalho. . .	67
5.2	Ilustração do funcionamento das redes autonômicas localizada e distribuída projetadas.	68
5.3	Fluxograma para regras de correlação relacionadas com reprogramação utilizadas nas simulações.	73
5.4	Fluxograma para regras de correlação relacionadas com reconfiguração de hardware utilizadas nas simulações.	74
5.5	Mensagens utilizadas pelos ESAs das redes simuladas.	75
6.1	Médias e desvios padrão do total de Bytes enviados pelos nós comuns.	82
6.2	Avaliação da produtividade na chegada de mensagens e dados relevantes no PA. .	83

6.3	Média e desvio padrão relativos às métricas de envio e perda de pacotes.	84
6.4	Médias e desvios padrão do total de bytes enviados pelos nós comuns. Resultados obtidos para os cenários 4 (autonômica distribuída) e 5 (autonômica localizada) contendo 200 nós sensores e variando a duração do intervalo de monitoração (50, 100 e 200 segundos).	88
6.5	Média e desvio padrão relativos às métricas de envio e perda de pacotes para os cenários 4 (autonômica distribuída), 5 (autonômica localizada) e 6 (não gerenciada). 89	
6.6	Médias e desvios padrão do número de dados e mensagens recebidas pelo PA e produtividade na recepção de dados. Resultados obtidos para os cenários 4 (autonômica distribuída) e 5 (autonômica localizada) contendo 200 nós sensores e variando a duração do intervalo de monitoração (50, 100 e 200 segundos).	90
6.7	Médias e desvios padrão do total de bytes enviados pelos nós comuns. Resultados obtidos para os cenários 4 (autonômica distribuída) e 5 (autonômica localizada), contendo 200 nós sensores e variando a quantidade de eventos na região monitorada (1, 2 e 4).	92
6.8	Média e desvio padrão relativos às métricas de envio e perda de pacotes para os cenários 4 (autonômica distribuída) e 5 (autonômica localizada).	93
6.9	Médias e desvios padrão do número de dados e mensagens recebidas pelo PA e produtividade na recepção de dados. Resultados obtidos para os cenários 4 (autonômica distribuída) e 5 (autonômica localizada), contendo 200 nós sensores e variando a quantidade de eventos na região monitorada (1, 2 e 4).	94
A.1	Fluxograma para regras de correlação utilizadas nos experimentos.	104

Lista de Tabelas

1.1	Comparativo entre diferentes plataformas de nós sensores.	5
5.1	Cenários implementados. A densidade das redes utilizadas foi sempre mantida em 1 nó/400m ²	77
6.1	Consumo médio total de energia dos nós comuns em Joules e porcentagem média do consumo por serviço (sensoriamento, comunicação e processamento).	81
6.2	Número de mensagens e dados enviados. Uma mensagem pode conter um ou mais dados sensorizados. Um dado relevante é aquele coletado durante a ocorrência de um evento.	82
6.3	Quantidade de memória RAM e ROM (em Bytes) necessária para cada um dos serviços (Monitoração, Análise, Planejamento e Execução) em conjunto com a BC, e para o ciclo autônomo em geral.	85
6.4	Consumo médio total de energia dos nós comuns em Joules e porcentagem média do consumo por serviço para os cenários 4 (autônomo distribuído) e 5 (autônomo localizado), contendo 200 nós e intervalo de monitoração variando entre 50, 100 e 200 segundos.	87
6.5	Número de mensagens e dados enviados. Resultados obtidos para os cenários 4 (autônoma distribuída) e 5 (autônoma localizada) contendo 200 nós e variando intervalo de monitoração (50, 100 e 200 segundos).	88
6.6	Consumo médio total de energia dos nós comuns em Joules e porcentagem média do consumo por serviço para os cenários 4 (autônomo distribuído) e 5 (autônomo localizado), contendo 200 nós e variando quantidade de eventos (1, 2 e 4).	91
6.7	Número de mensagens e dados enviados. Resultados obtidos para os cenários 4 (autônoma distribuída) e 5 (autônoma localizada), contendo 200 nós e variando a quantidade de eventos (1, 2 e 4).	92
6.8	Consumo médio total de energia dos nós comuns em Joules e porcentagem média do consumo por serviço para os cenários 4 (autônomo distribuído) e 5 (autônomo localizado), contendo 200 nós e variando o número de instruções para ciclo autônomo (40.000, 80.000 e 120.000 instruções).	95
6.9	Resultados das demais métricas para os cenários 4 (autônomo distribuído) e 5 (autônomo localizado), contendo 200 nós e variando o número de instruções para o ciclo autônomo (40.000, 80.000 e 120.000 instruções).	96

Capítulo 1

Introdução

Este trabalho propõe a aplicação do paradigma de computação autônoma no projeto de Redes de Sensores Sem Fio (RSSFs) capazes de se auto-gerenciar. As RSSFs, na maioria dos casos, são projetadas para operar em regiões remotas, onde a intervenção direta de técnicos e operadores é impraticável. Assim, essas redes precisam ser desenvolvidas a partir de hardware e software que permitam auto-gerenciamento. Nesse capítulo são introduzidos os principais conceitos relacionados com RSSFs (Seções 1.1 e 1.2) e computação autônoma (Seção 1.3), visto que esses são dois temas recentes de pesquisa importantes para o trabalho e que vem recebendo atenção da comunidade científica. Também são apresentados nas Seções 1.4, 1.5 e 1.6 as motivações, objetivos e contribuições desse trabalho, respectivamente. A organização do texto dessa dissertação está apresentada na Seção 1.7 e as considerações finais na Seção 1.8.

1.1 Redes de Sensores Sem Fio

Uma Rede de Sensores Sem Fio (RSSF) [Estrin et al., 2002, Ruiz et al., 2004] é considerada um tipo especial de rede ad hoc formada, na maioria dos casos, por centenas a milhares de elementos de rede, chamados nós sensores, depositados em ambientes remotos onde a manutenção por técnicos e a administração local podem ser impraticáveis. As RSSFs são projetadas para coletar dados de um ambiente, executar algum processamento sobre eles e entregar as informações obtidas para um observador externo através de um ou mais Pontos de Acesso (PAs). O objetivo dessas redes é o de monitorar e eventualmente controlar uma região de interesse. Existem diversas aplicações possíveis para as RSSFs em diferentes áreas, tais como monitoração ambiental, controle de poluição e níveis de gases tóxicos em centros urbanos e galpões industriais, monitoração de parâmetros vitais em um corpo humano, automação predial e industrial, exploração interplanetária, robótica, dentre outros. Todas as soluções propostas para uma RSSF devem considerar os aspectos específicos da aplicação para a qual a rede foi proposta. Uma solução, proposta para uma determinada RSSFs pode não ser interessante ou mesmo viável para outras com diferentes aplicações. Nesse caso, diz-se que as RSSFs são dependentes da aplicação. A Figura 1.1 ilustra de forma genérica RSSFs depositadas em determinada região de interesse, entregando dados para um Ponto de Acesso (PA) ou

observador externo, que é a entidade que possui interesse sobre os dados coletados pela rede. A Figura 1.2 apresenta algumas das aplicações para esse tipo de rede.

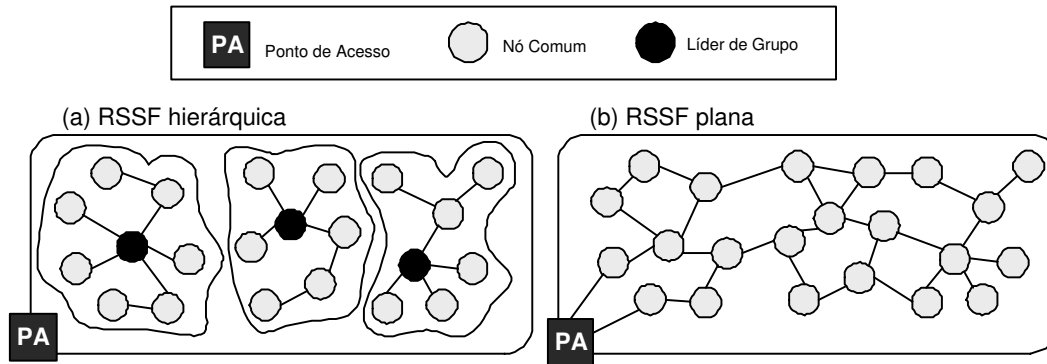


Figura 1.1: Rede de Sensores Sem Fio (RSSF).

Existem diferentes possibilidades de configurações para RSSFs, as quais definem que em sua composição, essas redes podem ser homogêneas (todos os componentes possuem mesma configuração de hardware) ou heterogênea (nós possuem diferentes capacidades de hardware). Em relação a organização, as RSSFs podem ser planas (sem formação de grupos) ou hierárquicas (nós sensores organizados em grupos, cada um deles com um líder), conforme ilustra a Figura 1.1. Os componentes das RSSFs podem ser estacionários ou móveis, estando distribuídos de forma regular (distribuição uniforme na área monitorada) ou irregular. Finalmente, a rede pode ser caracterizada como densa (grande quantidade de nós por unidade de área), balanceada (concentração e distribuição de nós considerada ideal) ou ainda esparsa (pequena quantidade de nós por unidade de área). Outras possibilidades de configurações para RSSF podem ser encontradas em [Ruiz, 2003].

As RSSFs possuem características particulares que as diferenciam em muitos aspectos das redes tradicionais. Dentre essas características pode-se citar a grande quantidade de elementos de rede, operação desacompanhada (sem manutenção e acompanhamento de técnicos e operadores), execução de uma aplicação comum por todos os nós sensores, deposição em ambientes inóspitos ou hostis, fluxo de dados predominantemente unidirecional, topologia dinâmica (nós podem ser descartados, perdidos, esgotarem suas fontes de energia, falharem permanente ou temporariamente, dentre outros fatores), grande dependência da aplicação para a qual foram projetadas, grande restrição de recursos, dentre outros [Ruiz, 2003]. As diferenças citadas acima apontam para a necessidade de se projetar algoritmos, protocolos, dentre outras soluções, específicas para as RSSFs, dados que aquelas existentes para as redes tradicionais não seriam adequadas na maior parte dos casos. Em particular, existirão diferenças significativas entre o gerenciamento desse tipo de rede e as redes tradicionais. Soluções de gerenciamento específicas, aderentes às características das RSSFs, seus componentes e aplicações são necessárias para se alcançar produtividade dos recursos e qualidade dos serviços providos.



Figura 1.2: Apresentação de alguns cenários de aplicações para RSSFs.

1.2 Nós Sensores Sem Fio

Uma RSSF é composta por um conjunto de dispositivos compactos chamados nós sensores. Esses nós executam essencialmente tarefas de sensoriamento do ambiente, processamento de dados e comunicação de informações. Esses dispositivos são frequentemente projetados com pequenas dimensões, o que faz com que os mesmos apresentem grandes restrições de recursos, tais como capacidade de processamento, armazenamento, comunicação e energia. Nesse caso, as soluções propostas para as RSSFs devem considerar tais restrições, consumindo a menor quantidade de recursos possível na execução de tarefas.

Os nós sensores são na maioria dos casos formados por uma unidade computacional, composta por memória e micro-controlador, uma fonte de bateria, uma unidade de transmissão e um ou mais dispositivos sensores, tais como unidade, temperatura e pressão (ver Figura 1.3). Em geral, baterias com capacidades finitas são utilizadas como fonte de energia para os nós sensores. A escolha da bateria a ser utilizada deve considerar algumas características como volume, condições de temperatura e capacidade inicial. Os tipos mais comuns de bateria para os nós sensores são linear simples, lítio NR e lítio *Coin Cell*. No entanto, ainda são possíveis fontes de energia solares, produzidas através de vibração, acústica e reações nucleares. A unidade de comunicação inclui todo o sistema de transmissão e recepção, amplificador e antena. Os dois tipos de comunicação mais utilizados nas arquiteturas de nós sensores são: (1) óptica, na qual o transmissor utiliza raios laser para envio de informação e (2) transmissão de ondas eletromagnéticas com frequências que variam de dezenas de KHz até GHz. O consumo de energia do transceptor é dependente da operação efetuada e tipicamente a transmissão de dados consome mais energia do que a recepção. A memória e o processador ou micro-controlador estão envolvidos nas atividades de computação realizadas pelo nó sensor. Quanto maior a frequência do processador, maior o consumo de energia. As operações de

leitura e escrita em memória também envolvem um consumo de energia que pode ser significativo de acordo com a frequência de uso e o número de ciclos de *clock* envolvidos. O processador normalmente opera em baixa frequência e a memória possui baixa capacidade de armazenamento. Um dispositivo sensor é um elemento que produz uma resposta mensurável para uma mudança na condição física (temperatura, pressão, campo magnético, estresse mecânico, presença ou ausência de movimento, áudio, vídeo, dentre outros). Esses dispositivos geralmente têm características físicas diferentes. Assim, numerosos modelos de nós sensores de complexidade variável podem ser construídos com base nas necessidades particulares das aplicações para a RSSF [Silva et al., 2004a].

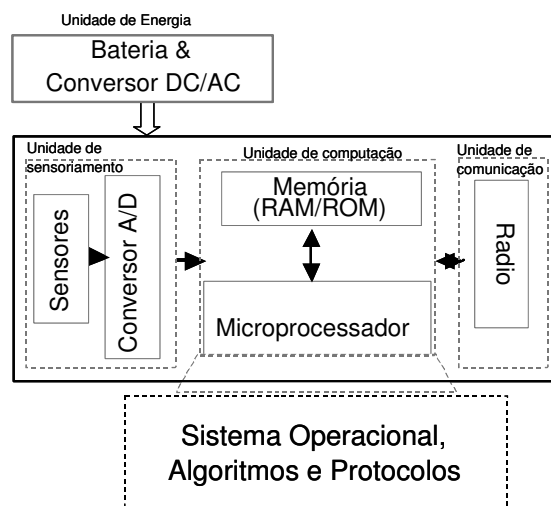


Figura 1.3: Principais componentes do nó sensor sem fio [Ruiz, 2003].

Apesar dos nós possuírem individualmente pouca capacidade computacional e de energia, um esforço colaborativo entre eles permite a realização de uma tarefa maior. Em alguns casos, uma RSSF pode ser composta de dispositivos atuadores que permitem ao sistema controlar parâmetros do ambiente monitorado.

Os nós sensores da RSSF são distribuídos por uma área e comunicam entre si formando uma rede ad hoc. Os nós sensores podem localmente processar dados coletados e enviá-los a um ou mais PAs. Esses nós também podem servir exclusivamente como roteadores das informações produzidas por outros nós comuns da rede.

Todos os componentes de um nó sensor necessitam de energia para operar. No entanto, esse recurso geralmente é oferecido por uma fonte pequena, finita e não recarregável. Além disso, a operação de substituição de fontes de energia pode ser difícil ou mesmo impraticável em certas aplicações. Assim sendo, a energia pode ser considerada o recurso mais importante e crítico para os elementos das RSSFs.

As restrições de tamanho e recursos, em particular a energia, também apresentam grande impacto em outro aspecto do projeto de nós sensores: a comunicação. Esses elementos em geral realizam envio de informações em múltiplos saltos, ou seja, os dados coletados são pas-

sados de um nó para outro, até que cheguem ao PA. Esse tipo de comunicação é empregada devido à impossibilidade de se utilizar rádios com dimensões significativas e que possibilitem um alcance maior, com melhor qualidade dos sinais transmitidos e consumo de energia compatível com as possibilidades dos nós sensores. O grande consumo de energia associado à comunicação [Hill e Culler, 2002], torna inviável para os nós sensores o envio de dados a grandes distâncias.

Atualmente existem algumas opções de plataformas de nós sensores desenvolvidas tanto por indústrias (Crossbow [Crossbow Inc., 2006] e MoteIV [MoteIV Corp., 2006], por exemplo) como em projetos científicos de diferentes universidades, tais como UCLA (*University of California Los Angeles*), Berkeley Univeristy, MIT (*Massachusetts Institute of Technology*), UFMG (Universidade Federal de Minas Gerais), dentre outras. Cada tipo de nó sensor possui características particulares para cada um de seus componentes, tais como diferentes tipos de fontes de energia, formas de comunicação, dispositivos sensores disponíveis, capacidade de memória e processamento, dimensões, peso e sistema operacional embutido. A Tabela 1.1 apresenta as principais características de nós sensores desenvolvidos por importantes projetos vinculados a empresas e universidades: WeC Motes [Hollar, 2000], Smart Dust [Warneke et al., 2001], μ AMPS [M. et al., 2000], WINS [P. e K., 2000], BSN [Node, 2006], ScatterWeb ESB [ScatterWeb, 2006], Telos [Crossbow Inc., 2006], TMote Sky [MoteIV Corp., 2006], TinyNode [Ferrière et al., 2006] e XYZ [Lymberopoulos e Savvides, 2005]. A arquitetura de nós sensores utilizada nesse trabalho, Mica Motes 2, será descrita em maiores detalhes na seção a seguir.

Tabela 1.1: Comparativo entre diferentes plataformas de nós sensores.

Nó Sensor	Processador	Taxa de Tx	Alcance
WeC Motes	Atmel ATmega128L@4MHz	10Kbps	20m
Smart Dust	Específico	10Kbps a 1Mbps	1 a 10Km
μ AMPS	StrongARM@133MHz	1Mbps	10 a 100Km
WINS	StrongARM@133MHz	100Kbps	100m
BSN	TI MSP430@8MHz	250Kbps	50m
ScatterWeb	TI MSP430@8MHz	19,2Kbps	100 a 300m
Telos	TI MSP430@8MHz	250Kbps	20 a 100m
TMote Sky	TI MSP430@8MHz	250Kbps	50 a 125m
TinyNode	TI MSP430@8MHz	1,2 a 152Kbps	1800m
XYZ	ARMTHUMB@57,6MHz	250Kbps	20 a 100m

1.2.1 Plataforma de Nós Sensores Sem Fio Mica Motes 2

A plataforma de nós sensores Mica Motes 2 foi inicialmente projetada por pesquisadores da universidade de Berkeley e atualmente é fabricada e vendida pela empresa *Crossbow Technology* [Crossbow Inc., 2006]. Existem disponíveis alguns tipos de nós sensores da família Mica Motes, dentre eles Mica Motes, Mica Motes 2, MicaDot e MicaZ. Além da plataforma computacional, a Crossbow também fabrica placas que contém vários dispositivos sensores

integrados, tais como sensores de temperatura, luminosidade, pressão barométrica, acelerômetro e magnetômetro, e também o elemento responsável por realizar a comunicação entre a rede e o observador, chamado de *Gateway*. A Figura 1.4 apresenta os nós sensores Mica2, MicaDot e o *Gateway*. As principais características particulares da plataforma Mica2, utilizada nesse trabalho, estão descritas a seguir:

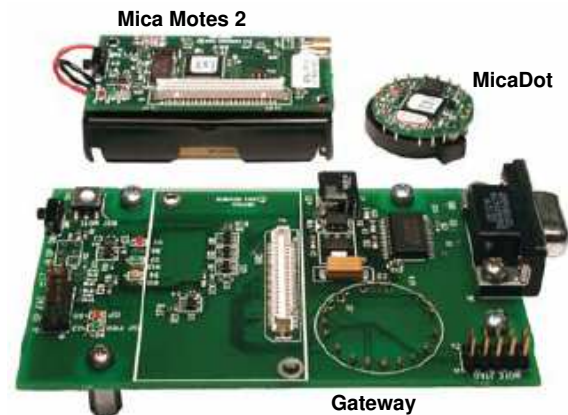


Figura 1.4: Ilustração de nós sensores da família Mica Motes.

- **Transceptor:** rádio transceptor CC1000 de 868/916MHz, 433MHz ou 315MHz, com taxa de transmissão de 19Kbps. A potência de transmissão pode ser regulada entre 0,01mW e 3,16mW, variando o alcance entre 5 e 91 metros. Consumo de 25mA para transmissão com máxima potência e 8mA para recepção. No modo *sleep* consome menos de $1\mu\text{A}$.
- **Bateria:** 2 baterias AA com tensão de operação 2.7-3.3V. O nó pode funcionar por 1 ano em modo de baixo consumo.
- **Sensores Disponíveis:** luminosidade, temperatura, acelerômetro, magnetômetro, ruído, umidade e pressão barométrica.
- **Processador:** micro-controlador de baixa potência Atmel Atmega 128L, que apresenta velocidade de 4MHz. Utiliza 8mA no modo ativo e menos que $15\mu\text{A}$ no modo *sleep*.
- **Memória:** memória de programa de 128KB, 512KB de memória não-volátil e 4KB de *Random Access Memory* (RAM).
- **Dimensões:** 5,71cm x 3,17cm x 0,63cm (aproximadamente $11,00\text{cm}^3$). O peso do nó Mica2 sem as baterias é de 18 gramas.
- **Características Especiais:** possui 51 pinos de expansão que permitem que outros elementos possam ser agregados ao sensor. A placa que possui os dispositivos sensores é agregada ao nó através destes pinos. Contém conversor analógico-digital de 10 bits e

3 Leds (vermelho, amarelo e verde) como interface para o usuário ou administrador da rede.

- **Sistema Operacional:** TinyOS [TinyOS, 2006]. TinyOS é o Sistema Operacional (SO) utilizado pelos nós sensores da família Mica Motes e também foi desenvolvido por pesquisadores da universidade de Berkeley. Ele é um SO de código aberto (*open-source*) projetado especificamente para RSSFs. Utiliza uma arquitetura baseada em componentes que permite rápida implementação enquanto minimiza o tamanho do código fonte, o que é uma característica desejável para elementos de redes que possuem severas restrições de memória. A biblioteca de componentes do TinyOS inclui protocolos de rede, serviços distribuídos, *drivers* para sensores e ferramentas para aquisição de dados. O modelo de execução dirigido a eventos do TinyOS permite gerenciamento de energia e flexibilidade programada, necessária para redes de comunicação sem fio e interfaces com o mundo físico com natureza imprevisível. Atualmente o TinyOS é utilizado por diversas outras plataformas além daquelas pertencentes a família Mica Motes, tais como Telos e TMote Sky.

1.3 Computação Autônômica

A Computação Autônômica define sistemas computacionais que são capazes de gerenciar a si próprios com nenhuma ou mínima intervenção humana [Kephart e Chess, 2003]. A inspiração para a criação do termo foi o sistema nervoso autônômico humano, o qual é capaz de manter o estado interno do corpo constante, prepará-lo para as tarefas do momento, cuidar de todas as funções vitais, dentre outras atividades, sem qualquer iniciativa ou esforço humano (ver Figura 1.5).

De forma similar ao que ocorre no corpo humano, a computação autônômica define um novo paradigma de controle e supervisão para sistemas computacionais, no qual os mesmos são capazes de realizar auto-gerenciamento, dados os objetivos de alto-nível definidos por seus administradores. O auto-gerenciamento é a essência da computação autônômica, e seu intuito principal é o de liberar os administradores de sistemas dos detalhes de operação e manutenção e prover aos usuários máquinas capazes de trabalhar com alto desempenho 7 dias da semana, 24 horas por dia [Kephart e Chess, 2003]. Dessa forma, os sistemas ficam encarregados de executarem as tarefas de gerenciamento, permitindo que gerentes humanos assumam uma posição de supervisão, podendo manter o foco no negócio, ou seja, nas atividades que são realmente importantes e fundamentais, e não no controle das tecnologias utilizadas. Um outro aspecto importante, que distingue sistemas autônomos ou automáticos de sistemas autônômicos é a possibilidade de aprendizado pelos elementos que compõem esses ambientes.

A computação autônômica também pode ser vista como um paradigma importante para o gerenciamento de sistemas para os quais a intervenção humana direta, através de gerentes, técnicos e operadores humanos pode ser difícil ou mesmo impossível, apresentando comportamento intrusivo proibitivo capaz de causar impacto negativo no desempenho dos mesmos.

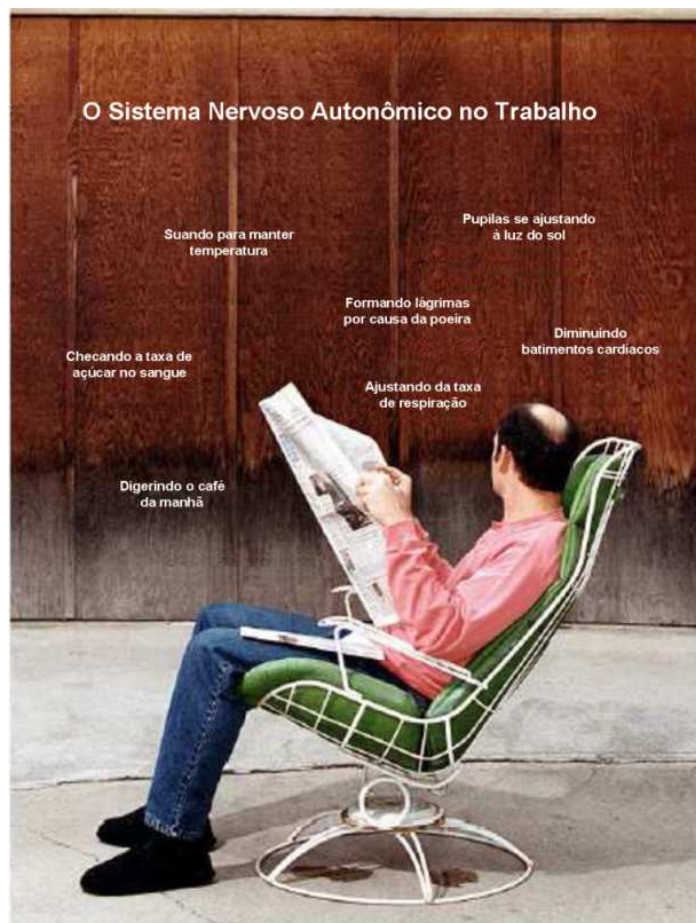


Figura 1.5: Atuação do sistema autônomo nervoso em seres humanos.

Conforme visto na Seção 1.1, as RSSFs representam sistemas desse tipo. Uma vez que paradigmas tradicionais que ainda mantém o foco na presença humana para execução de tarefas de controle e supervisão podem não ser adequadas, o uso da computação autônoma apresenta-se como uma opção interessante para esse tipo de rede. Esse trabalho propõe portanto o uso desse novo paradigma como forma de se gerenciar RSSFs, apresentando contribuições para o desenvolvimento dessa visão, propondo um modelo para se embutir comportamento autônomo nos elementos dessas redes.

Este novo paradigma de gerenciamento poderá ser aplicado a diversos contextos, desde simples componentes de hardware ou software até sistemas de Tecnologia da Informação (TI) de alta complexidade e redes de diversos tipos (com e sem fio, ad hoc e infra-estruturadas, locais ou globais, por exemplo). Neste caso, todos os serviços e características da computação autônoma podem ser igualmente implementados em todos estes ambientes, respeitando no entanto as necessidades e limitações de cada um deles. Para cada sistema computacional caberá uma instância de uma solução autônoma, capaz de interagir com outras soluções deste tipo e prover as melhores estratégias de controle e supervisão, dados os objetivos e particularidades da aplicação, componentes, estado atual dos ambientes interno e externo, dentre outros aspectos. Em última instância, a visão é que todos os sistemas computacio-

nais possuam alguma forma padronizada de controle autônomo, para que possam trocar recursos, serviços e dados, e ainda sejam formados por sub-sistemas autônomos e/ou façam parte de um sistema deste tipo mais amplo. A arquitetura de um sistema autônomo deve implementar quatro propriedades (também vistas como serviços) básicas: auto-configuração, auto-otimização, auto-cura e auto-proteção. A combinação dessas propriedades poderá permitir o desenvolvimento de outras funcionalidades importantes, tais como auto-organização, auto-sustento e auto-manutenção.

Desenvolver sistemas computacionais autônomos que apresentem a complexidade de TI embutida em sua estrutura, eliminando efetivamente sua visibilidade dos usuários finais, não é uma tarefa simples e trivial. A propósito, o fato de ser necessário um conjunto grande de tecnologias complexas para a construção de soluções autônomas é um paradoxo, uma vez que tais soluções são projetadas com o intuito de reduzir problemas ligados à complexidade de sistemas de TI.

Cada sistema autônomo deve possuir um Elemento Autônomo (EA) associado a ele (ver Seção 2.5 do Capítulo 2). Este elemento é a menor parte do sistema e deverá estar instalado em cada componente deste. Ele deverá implementar padrões para controlar os elementos gerenciáveis em questão, respeitando ao mesmo tempo os limites, características e objetivos dos mesmos. Os EAs devem gerenciar seus estados internos e seus relacionamentos com outros EAs, de acordo com políticas estabelecidas por humanos ou outros EAs. O auto-gerenciamento de um sistema autônomo será observado em geral pelo grande número de interações entre EAs, assim como pelo auto-gerenciamento interno que esses elementos realizam individualmente.

1.4 Motivações

O paradigma da computação autônoma terá um papel particularmente importante no gerenciamento de redes sem fio, dadas as características dos ambientes de operação, aplicações e dos próprios componentes desse tipo de rede. Considerando especialmente as RSSFs, o uso deste paradigma será em alguns casos a melhor forma de implementar e utilizar serviços e funções de gerenciamento. Conforme descrito na Seção 1.1, as RSSFs possuem diversas particularidades que as tornam diferentes dos outros tipos de redes. Esses aspectos específicos fazem com que as soluções de gerenciamento tradicionais, desenvolvidas de forma genérica com o intuito de serem aplicáveis a diversos contextos, tais como *Simple Network Management Protocol* (SNMP) e *Telecommunication Management Network* (TMN) por exemplo (ver Seção 3.5), possam não ser diretamente aplicáveis em função das restrições e particularidades apresentadas por essas redes. A quantidade de memória necessária e o consumo de energia com processamento e transmissão ou recepção de mensagens por exemplo, podem ser proibitivos para elementos de redes com tantas restrições como os nós sensores. A computação autônoma no entanto, prevê que os elementos do sistema devem gerenciar a si próprios de forma robusta, inteligente e eficiente. Nesse caso, os elementos autônomos e o modelo de interação entre eles serão desenvolvidos de acordo com as possibilidades e necessidades dos

elementos gerenciados. A visão da computação autônômica é portanto adequada às necessidades do gerenciamento de RSSFs, podendo prover os meios para que esse tipo de rede gerencie a si própria sem a necessidade de envolver operadores ou técnicos humanos. Desta forma, a RSSF passa a ser uma Rede Autônômica composta por nós sensores que possuem os serviços de um EA embutidos nos mesmos.

O uso da computação autônômica em RSSFs também facilitará a integração dessas com diversos outros tipos redes com e sem fios. Esse novo paradigma de gerenciamento prevê o uso de padrões abertos e comuns entre os diferentes sistemas e redes de forma a permitir coexistência e cooperação entre os mesmos. Uma RSSF autônômica cujos componentes contém EAs embutidos, seria facilmente integrada a outras redes autônômicas. Nesse caso, os elementos autônômicos das diferentes redes poderiam se comunicar através de interfaces padronizadas e bem definidas.

De acordo com o exposto até agora, verifica-se a necessidade de utilização de uma solução de gerenciamento para RSSFs e a aplicabilidade e aderência da computação autônômica nesse contexto. No entanto, até o momento não foram propostas soluções para a implementação integradas dos serviços de auto-gerenciamento nos elementos dessas redes. Existem trabalhos disponíveis na literatura que tratam de partes isoladas, tais como algoritmos e protocolos de auto-organização e auto-configuração para roteamento, controle de acesso ao meio, segurança, dentre outros aspectos. Uma vez que o desenvolvimento de uma RSSF autônômica pode demandar a implementação integrada de vários serviços, é importante a definição do modelo de um elemento autônômico que organize essa integração no nível de elemento de rede e permita a sua implementação em nós sensores sem fios, sendo aderente às características particulares desses elementos, da RSSF e de suas aplicações.

1.5 Objetivos

A construção de uma solução autônômica completa para qualquer tipo de sistema demanda a criação de um EA específico para o mesmo, o qual permitirá a execução dos serviços autônômicos, representação e gerenciamento eficiente dos recursos gerenciáveis e melhoria contínua dos serviços de controle e supervisão por meio das técnicas de aprendizado implementadas. O projeto de EAs particulares para cada dispositivo ou coleção de componentes pode ser considerado um dos principais desafios de pesquisa da computação autônômica [Kephart, 2005]. Este trabalho trata deste desafio: propor um elemento autônômico para RSSFs. É importante ressaltar que, no momento do desenvolvimento dessa dissertação, não foram encontrados disponíveis na literatura trabalhos que tratem da elaboração de um modelo de EA específico para RSSFs. Além disso, esse trabalho trata também do desafio da integração de duas tecnologias bastante recentes no meio acadêmico e industrial: RSSF e computação autônômica.

A definição dos serviços autônômicos que compõe o modelo de um EA e suas respectivas funções foi realizada através da observação das particularidades das RSSFs e seus componentes, conforme descrito nas Seções 1.1 e 1.2. Os aspectos e conceitos definidos pela computação autônômica foram utilizados para a definição de serviços e funções para o geren-

ciamento integrado de RSSFs em uma arquitetura específica para esse tipo de rede chamada Manna [Ruiz, 2003]. Complementando os objetivos dessa arquitetura, esse trabalho propõe um modelo de um elemento autônomo para os componentes desse tipo de rede, aderente às particularidades e características de hardware e software desses elementos, dos tipos de aplicações que utilizam RSSFs e expectativas do observador, que é a entidade que possui interesse sobre os dados coletados pela rede.

O formato genérico de um EA descrito em [Kephart e Chess, 2003] também foi utilizado como base para o modelo do elemento autônomo para RSSFs descrito nesse trabalho. Esse formato já prevê a estruturação de um EA como um ciclo composto por quatro etapas (monitoração, análise, planejamento e execução), as quais são chamadas de serviços nesse trabalho e uma base de conhecimento para armazenamento de informações. O EA é composto por um único gerente autônomo responsável por um ou mais elementos gerenciados. O formato define ainda dois pontos de contato entre gerente e elemento gerenciado: o sensor para coleta de dados durante a etapa de monitoração e atuador, que permite efetivação de modificações na etapa de execução. Interessante ressaltar que tal modelo pode ser aplicado em diferentes tipos de rede e sistemas, o que pode facilitar integração entre os mesmos.

1.6 Contribuições

As principais contribuições esperadas, considerando os objetivos apresentados na Seção 1.5 para o mesmo, estão descritas a seguir:

- Aplicação do paradigma de computação autônoma em RSSFs com objetivo de promover a produtividade dos recursos dessas redes e a qualidade dos serviços providos.
- Definição do modelo de um elemento autônomo, chamado Elemento Sensor Autônomo (ESA), a ser embutido nos elementos dessas redes, o qual poderá ser instanciado para diferentes plataformas de nós sensores.
- Avaliação do modelo proposto através da criação de uma instância particular para a plataforma de nós sensores Mica Motes 2. Os resultados obtidos comprovam os benefícios do uso da computação autônoma e a viabilidade de se embutir um ESA em elementos de uma RSSF.
- Extensão da arquitetura Manna para o gerenciamento integrado de RSSFs, provendo um ESA aderente à mesma, definido de acordo com suas características e propostas.

1.7 Organização

O restante desse texto está estruturado como a seguir: o Capítulo 2 apresenta o paradigma das redes autônomas e discute os principais aspectos relacionados com o mesmo, tais como níveis de autenticidade, serviços de auto-gerenciamento, definição de elementos autônomos e seus componentes e apresentação de uma proposta para desenvolvimento de soluções autônomas.

Além disso, o Capítulo descreve as necessidades e benefícios da aplicação dessa nova forma de gerenciamento em redes sem fio e em particular em RSSFs. O Capítulo 3 descreve os principais trabalhos relacionados encontrados na literatura, mostrando as contribuições de cada um deles e apontando as diferenças com relação as contribuições dessa dissertação. O modelo de um elemento autônomo específico para RSSFs está apresentado em detalhes no Capítulo 4. São discutidas no Capítulo as funções propostas para cada um dos serviços autônomos, os componentes da base de conhecimento e o funcionamento geral do ciclo autônomo proposto para esse tipo de rede. O projeto de simulação elaborado para avaliação do modelo proposto considerando uma plataforma de nós sensores específica e uma instância particular do modelo de ESA, e a análise dos resultados de desempenho obtidos estão descritos respectivamente nos Capítulos 5 e 6. O Capítulo 7 apresenta os comentários finais e também algumas direções para trabalhos futuros. O Apêndice A possui a descrição de experimentos desenvolvidos com o objetivo de realizar avaliação funcional do modelo de ESA proposto em ambientes reais.

1.8 Considerações Finais

Este capítulo apresentou brevemente os dois principais conceitos envolvidos nesse trabalho: RSSFs e computação autônoma. Para o primeiro conceito, foram apresentadas a definição, objetivo, características particulares e aplicações desse tipo de rede. Também se discutiu as características dos elementos que compõem uma RSSFs, apresentando cada um dos componentes desses elementos, suas funcionalidades e restrições. A plataforma Mica Motes 2 considerada de forma específica pra realização de simulações nesse trabalho, foi apresentada em maiores detalhes. Em relação ao segundo conceito, o capítulo apresentou a definição deste novo paradigma de gerenciamento, sua importância no cenário da indústria de TI atual e também qual foi a inspiração para o termo "computação autônoma". A importância desse paradigma em RSSFs também foi apontada. Além disso, o capítulo também apresentou as principais motivações, objetivos e contribuições esperadas com o desenvolvimento desse trabalho.

Capítulo 2

Redes Autônômicas

Este capítulo¹ aborda aspectos importantes relacionados com a computação autônômica, quais sejam definições e motivações (Seção 2.1), redes autônômicas sem fio (Seção 2.2), os níveis de autonomicidade (Seção 2.3), principais serviços autônômicos (Seção 2.4), definição e características de elementos autônômicos, ciclos de vida e relacionamento desses elementos (Seção 2.5) e um conjunto de passos propostos em um esquema para desenvolvimento de uma solução autônômica completa (Seção 2.6). Todos os aspectos citados acima são discutidos considerando-se a aplicação dos mesmos em ambientes de redes. Em particular, a abordagem desses aspectos para as RSSFs também é apresentada ao longo do capítulo, integrando assim esse tipo específico de rede à visão de gerenciamento provida pelo paradigma da computação autônômica. Ao final, a Seção 2.7 resume os conteúdos abordados nesse capítulo, ligando os mesmos aos objetivos desse trabalho.

A apresentação dos elementos mencionados acima para cada seção desse capítulo é importante para definição de conceitos relacionados com o principal objetivo desse trabalho: elaboração de um elemento autônômico para nós sensores sem fio. O capítulo apresentará a importância de implementação integrada de serviços autônômicos em redes, e em particular em redes sem fio e RSSFs, por meio da definição elementos autônômicos específicos, aderentes às características das redes, seus elementos, aplicações e objetivos. Essa implementação faz parte dos passos propostos para elaboração de uma solução autônômica completa para diversos tipos de redes, como as RSSFs, por exemplo.

2.1 Definição e Motivações

A implementação dos conceitos ligados à computação autônômica em redes de computadores leva a criação de um novo conceito: as redes autônômicas. Esse tipo de rede é capaz de realizar auto-gerenciamento de seus elementos e das conexões entre eles. Os serviços e funções de gerenciamento da rede são executados sem envolvimento de um gerente humano e de forma transparente para os usuários da mesma. Além disso, a rede é capaz de aprender com as

¹O conteúdo deste capítulo foi parcialmente publicado como minicurso no XXIV Simpósio Brasileiro de Redes de Computadores, 2006 [Braga et al., 2006a].

ações praticadas por seus elementos e análise dos resultados obtidos. A execução automática de tarefas e a possibilidade de aprendizado caracterizam o aspecto autônomo deste tipo de rede.

O desenvolvimento cada vez maior e mais acelerado de novas tecnologias no contexto das redes de computadores têm tornado esses ambientes computacionais bastante complexos e heterogêneos. Atualmente, além das redes tradicionais com fios, observamos o crescimento da disponibilização de outros tipos de redes, como as sem fio e *ad hoc* por exemplo. Observa-se inclusive a tendência em se unificar várias tecnologias (redes em malha) [Akyildiz e Wang, 2005] e disponibilizar conectividade e serviços para os usuários a todo tempo e em todo lugar (redes ubíquas) [Weatherall e Jones, 2002]. Imaginando os novos tipos de redes e serviços que são ou serão disponibilizados para usuários em um futuro próximo, verifica-se a necessidade do uso de técnicas de gerenciamento específicas e eficientes. Esse cenário de expansão tecnológica e aumento de complexidade, heterogeneidade, ubqüidade, conectividade e integração são alguns dos principais motivos que levam à necessidade de desenvolvimento de redes autônomicas, as quais devem utilizar novas funcionalidades, serviços e paradigmas de gerenciamento.

Assim como os mesmo algoritmos e protocolos podem não ser aplicáveis aos diferentes tipos de redes existentes, a solução de gerenciamento autônomo também deve ser desenvolvida considerando os aspectos particulares de cada uma delas. De fato, embora as soluções devam utilizar um padrão comum como base, cada uma delas trará implementações internas de forma diferenciada. O projeto de gerenciamento autônomo para uma rede metropolitana (*Metropolitan Area Network* - MAN) com fio, infra-estruturada e de propósito geral pode não ser o mesmo de uma rede pessoal (*Personal Area Network* - PAN) sem fio, *ad hoc* e de propósito específico, por exemplo.

2.2 Redes Autônomicas Sem Fio

Os serviços e funções de gerenciamento para redes que apresentam restrições de recursos, pouca tolerância ao *overhead* de comunicação e baixos níveis de qualidade de serviço, como as redes sem fio, devem ser implementados de forma cuidadosa, eficiente, robusta e transparente. Uma vez que as redes sem fio têm se tornado cada vez maiores, mais heterogêneas e abrangentes, com uma grande variedade de aplicações, tecnologias de hardware e software e usuários, estando presente não só em ambientes acadêmicos mas também em empresas e locais públicos, a demanda por eficiência no gerenciamento e qualidade de serviço nas transações se tornam cada dia maiores. Novos protocolos, serviços, sistemas operacionais e hardware têm sido desenvolvidos para estes ambientes em particular, e o movimento de integração entre as redes com e sem fio é cada vez mais significativo. Os usuários esperam que as redes sem fio apresentem níveis de eficiência iguais ou superiores aos das redes com fio, e que as mesmas estejam conectadas de uma forma transparente, aumentando assim sua utilidade, abrangência e complexidade.

Tipicamente, os elementos das redes sem fio possuem alguns tipos de restrição de recursos. O mais comum é a restrição de energia, devido a utilização de uma fonte finita desse

recurso. No entanto, ainda é possível observar restrições de memória, largura de banda, alcance, dentre outros. Desta forma, observa-se a necessidade de utilização de uma ferramenta de gerenciamento qualificada para as redes sem fio, capaz de controlar e resolver questões de compatibilidade, qualidade de serviço, utilização racional de recursos e conectividade.

As redes sem fio utilizam transmissão através de ondas eletromagnéticas para realização da comunicação de dados. Normalmente, dados digitais (seqüência de bits) ou analógicos (ondas mecânicas, por exemplo) são modulados em uma onda eletromagnética, conhecida como portadora, a qual possui as características necessárias ao meio de transmissão. Dentre os tipos de meios físicos para transmissão de dados sem fio estão a rádio-freqüência, micro-ondas e satélites. Uma vez que as ondas sofrem com vários tipos de efeitos físicos tais como interferência, atenuação e distorção que favorecem à perda dos dados modulados, as redes que utilizam este tipo de transmissão estão também sujeitas a baixos níveis de qualidade de serviço e produtividade de recursos, caso algum mecanismo de controle e supervisão não seja empregado. No entanto, este mecanismo deve ser desenvolvido de forma a não aumentar de forma significativa o *overhead* de comunicação da rede. Se a comunicação dos dados relevantes já não é uma tarefa simples, enviar além deles uma grande quantidade de dados de controle seria algo ainda mais complicado.

Além das características impostas a uma rede sem fio pelo tipo de meio físico utilizado, as tecnologias empregadas em suas camadas (e sub-camadas) de enlace, controle de acesso (MAC), rede e aplicação também influenciam diretamente em seu comportamento, capacidade, abrangência, dentre outros fatores. Pode-se citar como exemplo as diferenças observáveis entre redes que implementam tecnologia Bluetooth [Bluetooth, 2006], Wi-fi (*Wireless Fidelity*) [Wi-fi, 2006] e WiMax (*Worldwide Interoperability for Microwave Access*) [WiMax, 2006]. Os algoritmos de roteamento e endereçamento para essas redes também são diversos e diferenciados, além das diferentes possibilidades de aplicações.

Várias características de desempenho das redes sem fio, tais como alcance, taxa de dados, percentual de perda, *jitter*, dentre outros, são determinadas pelo protocolo de enlace utilizado e seus respectivos meios físicos [Wi-fi, 2006, WiMax, 2006, Bluetooth, 2006]. Neste caso, a solução autonômica não poderá negligenciar estes aspectos, devendo considerá-lo no modelo do elemento autonômico utilizado. Além disso, essas redes podem apresentar formas de comunicação entre seus elementos (topologias) diversas, tais como *ad hoc* e infra-estruturada. Todos esses componentes, que de uma forma ou de outra fazem parte do processo de comunicação em uma rede sem fio, devem ser conhecidos e utilizados no processo de definição e desenvolvimento da solução autonômica projetada para este tipo de rede. No projeto de EA para RSSFs proposto nesse trabalho, todos os dados que representam características da rede, seus elementos e sua aplicação estarão descritos em estruturas apropriadas utilizadas pelo mesmo para armazenamento de conhecimento.

A computação autonômica tem se apresentado como uma solução interessante de gerenciamento para as demandas por serviços e funções de controle e supervisão necessários para as redes sem fio. Esse tipo de rede requer soluções de gerenciamento que atuem de forma constante, competente, com alto índice de acertos e que seja capaz de correlacionar vários

parâmetros e possibilidades a cada tomada de decisão, sem levar a exaustão precoce dos recursos de seus elementos. Aspectos como complexidade, heterogeneidade e falta de mão-de-obra especializada, possuem grande impacto também para as redes sem fio, no sentido de dificultarem o desenvolvimento de soluções de gerenciamento interessantes para as mesmas.

Em especial, por se tratar de uma tecnologia nova, que ainda possui várias oportunidades de pesquisa a serem resolvidas e que deve se integrar a realidade das redes e sistemas já existentes, a questão do gerenciamento se torna ainda mais importante e complexa. Pode-se notar portanto a importância da implementação dos conceitos ligados as redes autonômicas no contexto das rede sem fio. Neste caso, tais redes passam a ser redes sem fio autonômicas, cujos componentes possuem um Elemento Autonômico (EA) embutido, capaz de gerenciá-los de forma organizada, seguindo os objetivos definidos para a rede e sua aplicação, utilizando para tal a menor quantidade de recursos possível e atendendo a todos os aspectos importantes mencionados acima.

Considerando especialmente as RSSFs, o uso deste paradigma será a melhor forma de implementar e utilizar serviços e funções de gerenciamento. A tarefa de se gerenciar RSSFs não é trivial, considerando que as essas redes apresentam todos os desafios descritos para as redes sem fio, além de outros ligados as redes *ad hoc* (organização e manutenção por exemplo), redes móveis e aqueles relacionados às suas características particulares (ver Seção 1.1). A implementação de recursos autonômicos nessas redes será importante para a realização de controle, supervisão e manutenção dos parâmetros de qualidade de serviço adequados e deve ocorrer de forma gradual, obedecendo a certos níveis de autonomicidade, conforme mostra a Seção 2.3 a seguir. Desta forma, a RSSF passa a ser uma rede autonômica composta por nós sensores que possuem os serviços de um EA.

2.3 Níveis de Autonomicidade

A evolução das redes autonômicas deve ser gradativa. Os desafios de pesquisa envolvidos com esse paradigma são muitos, e todos devem ser superados antes que uma solução puramente autonômica seja desenvolvida. Desta forma, é esperado que as redes passem por diferentes níveis, os quais indicam o quão autonômica é a solução de gerenciamento utilizada. Existem cinco níveis definidos para a classificação de redes autonômicas [Kephart e Chess, 2003]. A Figura 2.1 descreve as principais características de cada um destes níveis.

(1) Nível Básico Neste nível, cada equipamento é gerenciado independentemente de acordo com seu fabricante. O processo de gerenciamento é informal, reativo e manual, e as ferramentas utilizadas são locais e específicas para cada produto/plataforma.

(2) Nível Gerenciado No segundo nível são utilizadas tecnologias de gerenciamento de sistemas apenas para a coleta de informações. O processo de gerenciamento é documentado e melhorado através do tempo, mas ainda manual.

NÍVEL 1 BÁSICO	NÍVEL 2 GERENCIADO	NÍVEL 3 PREDITIVO	NÍVEL 4 ADAPTATIVO	NÍVEL 5 AUTÔNOMICO
<ul style="list-style-type: none"> MÚLTIPLAS FONTES DE GERAÇÃO DE DADOS NO SISTEMA REQUER EQUIPE DE TI GRANDE E ALTAMENTE HÁBIL 	<ul style="list-style-type: none"> CONSOLIDAÇÃO DOS DADOS ATRAVÉS DE FERRAMENTAS DE GERENCIAMENTO EQUIPE DE TI ANALIZA E TOMA AÇÕES 	<ul style="list-style-type: none"> SISTEMA MONITORA, CORRELACIONA E RECOMENDA AÇÕES EQUIPE DE TI APROVA E INICIA AÇÕES 	<ul style="list-style-type: none"> SISTEMA MONITORA, CORRELACIONA E TOMA AÇÕES EQUIPE DE TI GERENCIA PERFORMANCE FRENTE AO ACORDO DO NÍVEL DE SERVIÇO 	<ul style="list-style-type: none"> COMPONENTES INTEGRADOS DINAMICAMENTE GERENCIADOS POR POLÍTICAS/REGRAS DE NEGÓCIO EQUIPE DE TI FOCA NA CAPACITAÇÃO DAS NECESSIDADES DO NEGÓCIO
	<ul style="list-style-type: none"> MAIOR CONHECIMENTO DO SISTEMA PRODUTIVIDADE MELHORADA 	<ul style="list-style-type: none"> DEPENDÊNCIA EM GRANDES HABILIDADES DA EQUIPE DE TI REDUZIDA TOMADA DE DECISÃO MAIS RÁPIDA E MELHOR 	<ul style="list-style-type: none"> AGILIDADE E ELASTICIDADE DE TI COM A MÍNIMA INTERAÇÃO HUMANA 	<ul style="list-style-type: none"> POLÍTICAS DE NEGÓCIO DIRIGEM O GERENCIAMENTO DE TI AGILIDADE E ELASTICIDADE DO NEGÓCIO
MANUAL		AUTÔNOMICO		

Fonte: IBM Global Services and Autonomic Computing, IBM White Paper (Outubro 2002).

Figura 2.1: Classificação indicativa do nível de autonomicidade de uma rede [Ganek e Corbi, 2003].

(3) Nível Preditivo No nível Preditivo, pode-se observar o uso de técnicas de correlação para vários elementos de uma mesma rede, permitindo reconhecimento de padrões, previsão de um modelo ótimo de configuração e indicação de ações a serem tomadas pelo administrador. O processo no nível preditivo é pró-ativo.

(4) Nível Adaptativo Neste nível os componentes de rede podem tomar as ações corretas automaticamente, baseados nas informações disponíveis e no conhecimento gerado por eles próprios sobre o que está acontecendo na rede e no ambiente que a cerca. A operação dos elementos é guiada por SLAs (*Service Level Agreements*) definidas pelos administradores humanos. O processo de gerenciamento inclui automação das melhores práticas de gerenciamento de recursos e transações.

(5) Nível Autônomico Este é o nível mais avançado, no qual a operação da rede é governada por políticas e objetivos de negócio. Os usuários e administradores apenas monitoram o processo e/ou alteram os objetivos. Todas as melhores práticas de gerenciamento de serviços e recursos de TI são automatizadas.

Em geral, mesmo as redes e sistemas de TI tradicionais, em sua maior parte podem ser consideradas de nível básico ou gerenciado, apresentando nenhuma ou poucas práticas de gerenciamento, realizado por gerentes humanos de forma isolada e manual. No entanto, para esses ambientes já existem soluções isoladas ou mesmo parcialmente integradas que permitem aos mesmos apresentarem características dos níveis preditivo e até mesmo adaptativo. Por outro lado, considerando-se as soluções de gerenciamento para redes sem fio, pouco se tem avançado no sentido de propor alternativas autônomicas com características de níveis mais avançados. Em particular, para as RSSFs existem na literatura trabalhos que tratam de

serviços de gerenciamento apenas de forma isolada. Para avançar no nível de autenticidade das soluções de gerenciamento para RSSFs, é importante o uso de um elemento autônomo (ver Seção 2.5) específico, que integre serviços e funções, execute algoritmos de aprendizado e construa uma base de conhecimento completa e robusta.

A implementação integrada de diversos serviços de auto-gerenciamento por meio de um elemento autônomo e o uso de técnicas ligadas ao aprendizado de máquina, permitirão a evolução dos sistemas autônicos até o nível 5, ou seja, sistemas totalmente autônicos. A próxima seção apresenta a definição de alguns dos importantes serviços de auto-gerenciamento.

2.4 Serviços de Auto-gerenciamento

As redes autônicas devem implementar uma série de serviços de auto-gerenciamento para assegurar que suas tarefas serão realizadas de forma transparente, contínua e contando com pouca ou nenhuma intervenção humana. Existem quatro serviços básicos que todo sistema autônomo deve implementar: auto-configuração, auto-otimização, auto-cura e auto-proteção (ver Figura 2.2) [Ruiz, 2003, Ganek e Corbi, 2003].

- **Auto-configuração:** as redes autônicas deverão se configurar e reconfigurar automaticamente de acordo com políticas de negócio fornecidas por seus administradores. Políticas de negócio são aquelas que definem o que deve ser feito e não como fazer. A rede deve ser adaptar à adição de novos componentes, software ou mesmo políticas. Em RSSFs parâmetros de configuração são modificados dinamicamente de acordo com as condições variáveis e até mesmo imprevisíveis do ambiente monitorado. Em nível de elemento de rede, pode-se citar a modificação de estados dos componentes dos nós sensores, tais como processador e dispositivo de comunicação, como exemplos de auto-configuração em RSSFs.
- **Auto-otimização:** as redes autônicas devem buscar continuamente melhorar sua operação, identificando e analisando oportunidades de se tornarem mais eficientes em desempenho e custo. Em RSSFs todas as ações de otimização devem ser executadas considerando-se o compromisso de custo-benefício, ou seja, a operação de otimização deve ser realmente considerada necessária e compensada pelo aumento de precisão ou diminuição no consumo de algum recurso.
- **Auto-cura:** pode-se dizer que a auto-cura é formada por dois outros serviços: auto-diagnóstico e auto-recuperação. Utilizando conhecimento sobre a configuração da rede, um componente de diagnóstico de problemas deve analisar informações em arquivos de log, possivelmente auxiliado por dados de monitorações adicionais requisitadas por ele. Em RSSFs, os gerentes autônicos devem realizar auto-cura sempre que necessário, respeitando também a quantidade de recursos disponíveis para tal. A descoberta de um problema e sua correção podem levar a uma melhoria no comportamento do nó sensor, maior precisão e menor consumo, compensando assim as ações de gerenciamento.

- **Auto-proteção:** as redes autonômicas deverão ser especialistas em auto-proteção. Elas se defenderão em dois sentidos: (i) defesa do sistema como um todo contra problemas de larga escala e correlacionados, causados por ataques maliciosos e (ii) falhas isoladas ou em cascata que permanecem incorretas. Uma vez que é comum a ocorrência de falhas em RSSFs, os componentes dessas redes devem estar preparados para diferenciá-las entre intencionais e não-intencionais, resguardando dados coletados de possíveis intrusos e garantindo o funcionamento de parâmetros mínimos de qualidade de serviço da rede como um todo. Medidas de auto-proteção tais como detecção de intrusos e autenticação, podem ou não ser proibitivas de acordo com a quantidade de recursos necessários e disponíveis. Mais uma vez o compromisso entre precisão e consumo deve ser considerado.



Figura 2.2: Principais serviços autonômicos: auto-configuração, auto-otimização, auto-proteção, auto-cura.

Além dos quatro serviços fundamentais descritos acima, outros poderão ser implementados em redes autonômicas, inclusive através da combinação entre os mesmos e/ou com aqueles descritos acima. Esses demais serviços também são importantes para uma solução autonômica e devem ser implementados por meio do elemento autonômico responsável. A seguir alguns desses serviços estão descritos em mais detalhes:

- **Auto-conhecimento:** para gerenciar componentes e enlaces entres eles de acordo com os princípios da computação autonômica, uma rede deste tipo deve conhecer a si própria e os elementos que a compõe. Este serviço fornece as bases para a execução de praticamente todos os demais serviços das redes autonômicas. O auto-conhecimento é particularmente importante em RSSFs, dada as restrições de recursos que os elementos dessas redes possuem. É fundamental que os nós sensores tenham uma conhecimento mais preciso possível de seus recursos, para que possam realizar utilização otimizada e precisa, prolongando o tempo de vida da rede e possibilitando a execução de outros serviços de gerenciamento.
- **Auto-consciência:** além do conhecimento dos recursos próprios, cada elemento autonômico presente em uma rede autonômica deve conhecer as características do ambiente que o cerca. Em RSSFs a interação entre os elementos da rede é normalmente

intensa, visto que na maior parte das vezes os nós se comunicam utilizando esquema de múltiplos-saltos. Assim, o conhecimento a cerca do ambiente, dos vizinhos e de suas ligações com os mesmos pode levar a um comportamento mais preciso e um gerenciamento mais eficiente.

- **Auto-aprendizado:** a característica marcante que distingue sistemas ou redes autônomos daqueles autônomicos é justamente a capacidade de aprendizado a partir das experiências e resultados obtidos em ações anteriores. Existem várias maneiras de se aplicar técnicas de aprendizado de máquinas em sistemas computacionais [Mitchell, 1997], e todas elas devem ser cuidadosamente testadas para o contexto das redes autônomicas. Assim como para qualquer tipo de rede, ações de auto-aprendizado são importantes em RSSFs. No entanto, a restrição de recursos desse tipo de rede demanda uma atenção especial no uso de técnicas de aprendizado, visto que as mesmas podem ser complexas, exigindo grande quantidade de processamento, memória e conseqüentemente de energia. O elemento autônomico deve saber escolher os algoritmos apropriados e os momentos em que deve executá-los ou não.
- **Auto-diagnóstico:** este serviço pode ser entendido como uma parte da auto-cura. Diferenciar os pontos problemáticos daqueles que estão apenas apresentando problemas colaterais, é o grande desafio deste serviço. O auto-diagnóstico em RSSFs pode ser feito em diferentes níveis de gerenciamento, possuindo diversas funções tais como identificação de problemas nos componentes do próprio nó sensor (memória, dispositivos sensors, micro-controlador, dentre outros), localização de elementos de redes defeituosos ou indisponíveis na rede, identificação de não-cumprimento de níveis de qualidade de serviço definidos pelas regras de negócio.
- **Auto-recuperação:** a auto-recuperação também é uma das partes que compõe a auto-cura. Uma vez diagnosticado e isolado o ponto gerador de problemas em uma rede autônomicas, uma ou mais ações de recuperação do sistema deve ser efetuadas. Para cada problema identificado por uma RSSF por meio de seu serviço de auto-diagnóstico deve corresponder uma ou mais ações de auto-recuperação. No entanto, as ações não devem ser executadas sem a verificação do compromisso entre precisão ou necessidade e consumo de recursos. Em determinadas circunstâncias, pode ser mais indicado que um nó não execute ações de auto-recuperação devido a indisponibilidade de recursos necessários.
- **Auto-organização:** em muitos casos, uma rede necessitará organizar e reorganizar seus elementos e os enlaces entre estes para a realização de suas tarefas. Em RSSFs esse serviço é utilizado para se alcançar estruturas organizacionais necessárias sem a requisição de intervenção humana. A eficiência desse processo organizacional pode ser dependente da deposição da rede e o grau e precisão das informações disponíveis para nós sensores. Exemplos de aplicação desse serviço nessas redes são a construção de grupos e da infra-estrutura de roteamento.

- **Auto-serviço:** em muitos casos as redes autonômicas devem ser auto-suficientes no sentido de serem provedoras e usuárias de seus próprios serviços. Uma RSSF é usuária de seus próprios serviços, dados que os elementos da rede são responsáveis por produzir e entregar dados ao PA. Assim, essa rede deve prover serviços de disseminação, sensoriamento e processamento, antecipando recursos necessários enquanto mantém a complexidade escondida.
- **Auto-sustento:** Uma vez implementados os serviços de auto-conhecimento e auto-consciência, uma rede autonômica deve ser capaz de verificar e dimensionar suas necessidades em relação aos recursos utilizados, tais como energia, memória, capacidade computacional, dentre outros. Essa capacidade é chamada de auto-sustento, e permite que os elementos autonômicos da rede planejem seus esquemas de auto-organização, auto-configuração, auto-serviço e auto-otimização de maneira mais ajustada, específica e eficiente. Em última instância, os recursos disponibilizados pelos elementos gerenciados serão utilizados pelos gerentes autonômicos de forma racional, sem desperdícios. Em RSSFs o auto-sustento em termos do consumo de energia pode ser considerado um dos mais importantes, evitando desperdício de um recurso essencial para o funcionamento dos elementos dessas redes.
- **Auto-manutenção:** os serviços de auto-configuração, auto-otimização e auto-sustento estão relacionados com a auto-manutenção. Neste serviço a rede como um todo e cada um de seus elementos deve ajustar parâmetros e configurações no intuito de alcançar ou manter objetivos pré-determinados. Existem alguns importantes exemplos de aplicação de auto-manutenção para RSSFs, tais como manutenção da área de cobertura (rede identifica e retira de serviço temporariamente nós redundantes, para manter por mais tempo a cobertura da rede, em termos de sensoriamento e/ou comunicação), gerenciamento de mobilidade, manutenção de qualidade de serviços, dentre outros.

A expectativa é que as redes sem fio, e em especial as RSSFs venham a ser desenvolvidas considerando-se todo o conjunto de serviços autonômicos definidos acima. Para tanto, os componentes dessas redes devem possuir elementos autonômicos embutidos, capazes de permitir a realização integrada das tarefas de cada serviço. A Seção seguinte descreve o que são os elementos autonômicos, seus componentes, ciclos de vida e de relacionamento.

2.5 Elemento Autonômico (EA)

Essa seção apresenta conceitos e características de elementos autonômicos de uma forma genérica. A definição desses conceitos no contexto específico de RSSFs é a principal contribuição desse trabalho, e está apresentada no Capítulo 4.

Um Elemento Autonômico (EA) é a menor parte de um ambiente autonômico e podem ser vistos como sistemas individuais que contém recursos e provêm serviços para humanos e/ou outros EAs (ver Figura 2.4). Em [Kephart e Chess, 2003] está descrito um modelo genérico

para definição de um EA. Esse modelo propõe a formação de um EA através de um único gerente autonômico e um ou mais elementos gerenciados. Além disso define etapas a serem executadas pelo gerente de forma seqüencial e contínua (laço ou ciclo autonômico de controle infinito) e uma base de conhecimento utilizada pelo mesmo para recuperação e armazenamento de diversas informações. As funções que compõem cada uma das etapas do laço de controle contínuo de um EA, sua base de conhecimento, as etapas presentes em seu ciclo de vida e a forma como elas são executadas, bem como o modelo de relacionamentos entre EAs de uma rede autonômica, devem ser definidos de forma específica e particular para cada tecnologia e tipo de rede ou sistema.

2.5.1 Elementos Gerenciados

É possível se ter EAs embutidos em diferentes tipos de elementos gerenciados, como dispositivos de hardware (discos e CPUs por exemplo), unidades de software e até mesmo em coleções de componentes, como um computador pessoal. Em particular nas redes autonômicas, os componentes (elementos de rede) são vistos como a menor parte do sistema, ou seja, são os elementos gerenciados. O EA embutido em cada um desses componentes é capaz de monitorá-los e controlá-los, implementando serviços e funções de gerenciamento. Vale lembrar que o hardware e/ou software de um componente da rede, apesar de não serem o foco específico desse tipo de ambiente, podem ser sistemas autonômicos independentes, possuindo capacidade de auto-gerenciamento. Os elementos gerenciáveis (Figura 2.3) possuem interfaces gerenciáveis, que permitem que gerentes autonômicos possam receber informações e atuar sobre os mesmos. Uma vez que diferentes elementos podem apresentar diferentes formas específicas de interação, é interessante a implementação de interfaces padronizadas, facilitando o contato entre gerentes e diversos elementos gerenciados. A implementação dessas interfaces é chamada de *touchpoint*, sendo ela composta por duas partes: o sensor, que é o mecanismo que permite obtenção de detalhes dos elementos gerenciáveis, e os atuadores, responsáveis pelos mecanismos de mudança de comportamento dos mesmos. Neste trabalho, consideramos que os componentes gerenciáveis dos nós sensores, componentes das RSSFs, possuem interfaces desses tipo (*touchpoints*) implementadas, ou seja, não está no escopo da dissertação o desenvolvimento das mesmas [AC Blueprint, 2006].

2.5.2 Gerente Autonômico

Um EA contém um único gerente autonômico que representa e monitora o(s) elemento(s) gerenciado(s) (Figura 2.4). Nesse caso, cada EA atua como um gerente, responsável por promover a produtividade dos recursos e a qualidade dos serviços providos pelo componente da rede no qual está instalado. O gerente executa um laço de controle contínuo que realiza monitoração e análise de dados internos e externos ao elemento gerenciado, e também planejamento e execução de ações corretivas ou que possam aprimorar o funcionamento deste elemento. Além disso, o EA possui um módulo que é uma base de conhecimento onde armazena dados, regras, limites, dentre outros valores e informações. Assim como os elementos gerenciados, os gerentes

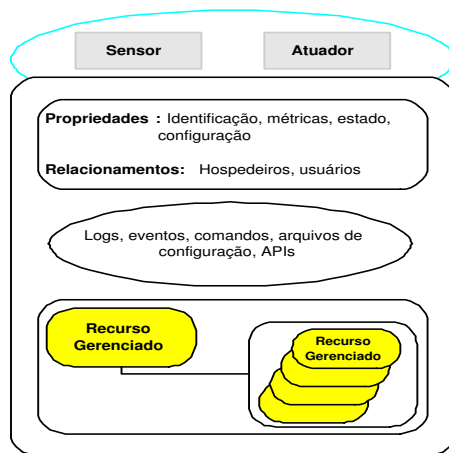


Figura 2.3: Estrutura Interna de um elemento gerenciado [AC Blueprint, 2006].

autonômicos também possuem interfaces de sensor e atuador (*touchpoints*) para comunicação com outros gerentes autonômicos e outros componentes da infraestrutura distribuída da rede. Com a utilização dessas interfaces, vários gerentes podem colaborar para o gerenciamento como um todo da rede ou sistema, sendo cada gerente responsável por um escopo de recursos particular e atuando em algum papel no gerenciamento global [AC Blueprint, 2006]. Também não está no escopo desse trabalho a definição dos *touchpoints* para interação entre os elementos autonômicos embutidos nos nós sensores de uma RSSF.

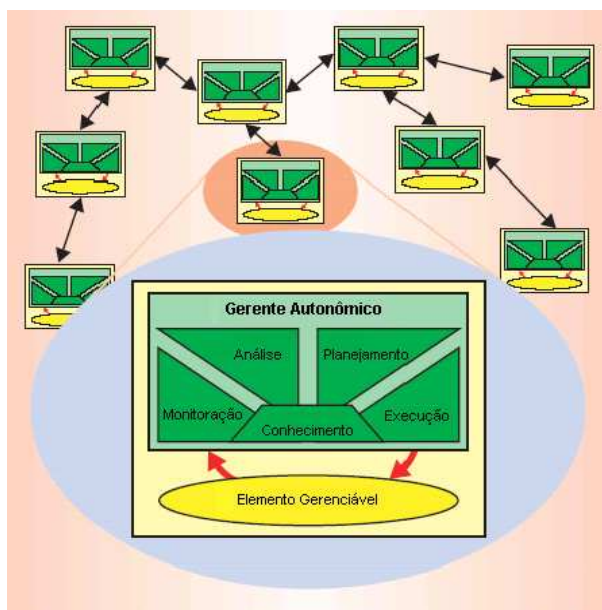


Figura 2.4: Estrutura Interna de um Elemento Autonômico [Kephart e Chess, 2003].

2.5.3 Serviços de um Gerente Autônomo

A definição da arquitetura de elementos autônômicos não contém as funções e escolhas de implementação específicas para a estrutura interna de tarefas de um gerente autônomo. No entanto a arquitetura organiza a estrutura interna do mesmo em um conjunto de capacidades ou etapas, chamadas nesse trabalho de serviços. A seguir são apresentados de forma geral cada um desses serviços. A principal contribuição desse trabalho (ver Capítulo 4) é a definição das funções e escolhas de implementação para cada serviço, considerando-se especificamente as RSSFs.

2.5.3.1 Serviço de Monitoração

O auto-conhecimento e a auto-consciência (ver Seção 2.4) são conceitos importantes e devem ser providos pelo modelo de elemento autônomo utilizado. O serviço de monitoração de um elemento autônomo é responsável pela implementação dos dois conceitos acima. A coleta de parâmetros internos e externos aos elementos gerenciados deve portanto ser realizada por esse serviço. A monitoração pode ser projetada de diversas maneiras, dependendo do sistema autônomo ao qual pertence. Em redes tradicionais por exemplo, vários parâmetros de hardware, software e do ambiente podem ser monitorados e em seguida combinados e armazenados na base de conhecimento (ver Seção 4.6). Além disso, a frequência de monitoração pode ser alta, uma vez que computadores tradicionais podem processar de forma eficiente mais de uma tarefa ou processo. Além do acesso direto as informações internas, as quais podem ser providas de uma forma padronizada por meio dos *touchpoints*, o EA pode obter informações externas com o recebimento de mensagens de gerenciamento ou monitoração do fluxo de dados da rede. Os dados obtidos com a monitoração podem ainda ser classificados como estáticos e dinâmicos, sendo importante a realização da monitoração periódica com frequência adequada para se manter valores atualizados desses últimos.

Monitoração de Fluxos de Dados: a operação de monitoração de fluxos por EAs é uma forma importante de aquisição de dados externos, ou seja, dados que refletem o estado da rede e da vizinhança (auto-consciência). Segundo [Babcock et al., 2002], fluxos de dados são transientes e seus ítems de dados podem ser representados por tuplas relacionais. Os fluxos de dados tipicamente apresentam chegada contínua, múltipla, rápida e variável de dados. Fluxos de dados diferem do modelo de armazenamento relacional convencional em vários aspectos, tais como a chegada de elementos (dados) do fluxo em tempo real, a falta de controle sobre a ordem em que os elementos de dados são processados e a potencial falta de limites de tamanho ou quantidade. Uma vez que um elemento de um fluxo de dados foi processado, ele deve ser descartado ou arquivado. EAs em redes de qualquer tipo podem realizar monitoração do meio de comunicação por onde passam os fluxos de dados enviados por todos os elementos da rede. Essa operação permite que diversos parâmetros relacionados com a rede, o ambiente e a vizinhança sejam monitorados.

2.5.3.2 Serviço de Análise

A verificação dos contextos interno e externo de um elemento autônomo será realizada pelo serviço de análise. É responsabilidade deste serviço transformar os dados de entrada, obtidos pelo serviço de monitoração, em informações úteis que possam ser analisadas, e que levem a conclusões sobre diversos aspectos, como desempenho ou falhas por exemplo, ligados ao elemento gerenciado. O serviço autônomo implementado por esta etapa é o auto-diagnóstico. O EA deve ser capaz de entender o que está acontecendo com ele próprio e com o ambiente a sua volta (elementos vizinhos e as ligações entre eles). Os dados recém coletados pelo serviço de monitoração devem ser anexados e comparados aos outros previamente obtidos, e uma correlação de vários fatores realizada. O resultado produzido por este serviço, ou seja, o diagnóstico feito pelo EA a respeito dos elementos gerenciados e de sua vizinhança, influenciará de forma significativa no desempenho do sistema autônomo, pois um conjunto de ações será definido e executado utilizando-o como base. Dentre as possíveis técnicas para realização de análise de dados estão a mineração de dados e correlação de dados ou eventos. Uma outra possibilidade é a realização de análise estatística de um volume de dados.

Mineração de Dados: essa técnica, também conhecida como “descoberta de conhecimento em bancos de dados” pode ser definida como um processo não trivial de extração de informações implícitas, previamente desconhecidas e potencialmente úteis, tais como regras de conhecimento, limites e regularidades, a partir de dados em bancos de dados [Chen et al., 1996]. Existem atualmente vários requisitos e desafios na implementação de soluções de mineração de dados, dentre elas, tratamento de diferentes tipos de dados, eficiência e escalabilidade dos algoritmos, utilidade, confiabilidade e expressividade dos resultados da mineração, segurança (proteção e privacidade), tratamento de dados provenientes de várias fontes, dentre outros.

Correlação de Eventos: o objetivo de um algoritmo de correlação é reduzir o número e enriquecer o significado dos eventos observados em uma dada rede ou sistema. Existem vários métodos e algoritmos disponíveis para realização de correlações, tais como correlação baseada em regras, lógica difusa, redes bayesianas, raciocínio baseado em modelos, quadro negro, filtragem, correlação por votação, redes neurais artificiais, dentre outros [Meira, 1997]. O uso de um ou outro método em uma rede autônoma depende de fatores tais como, capacidade computacional (processamento e memória) dos elementos da rede, quantidade de energia disponível (se aplicável), nível de precisão desejado, dentre outros fatores. Os EAs podem ainda optar por manter opções de algoritmos de correlação armazenados, ou mesmo solicitar a uma fonte externa novos métodos, utilizando-os de acordo com cada situação em particular.

Análise Estatística: o principal papel dessa análise é estabelecer de forma objetiva a relevância estatística dos resultados obtidos, de acordo com limites pré-estabelecidos. Para realizar análise estatística, o serviço de análise deve utilizar/determinar, dentre outros itens,

objetivos da análise, testes mais adequados, hipóteses estatísticas, variáveis a serem estudadas e dados individuais utilizados como valores de entrada. A análise estatística pode ser realizada por meio do teste de hipóteses e/ou estimando-se probabilidades [L.C. e A.A., 2001].

2.5.3.3 Serviço de Planejamento

O serviço de planejamento de um EA possui como principal objetivo escolher ou elaborar um plano de ações a serem seguidas, de acordo com o resultado da análise realizada pelo serviço anterior (Seção 4.3). Mais uma vez, um conjunto de funções associadas a este serviço pode ser definido de acordo com o grau de precisão e autenticidade desejado e o nível de complexidade permitido. O EA pode implementar conceitos bastante sofisticados de aprendizado de máquina por exemplo, caso ele possua recursos computacionais compatíveis. Alguns serviços autonômicos podem estar relacionado a esta etapa, de acordo com as ações que compõem o plano elaborado. Dentre eles pode-se citar o auto-serviço, auto-sustento, auto-manutenção, auto-organização, dentre outros. Uma ou mais ações do plano podem estar ligadas a um ou mais serviços autonômicos. O serviço de planejamento deve inclusive definir qual será a ordem de execução das ações do plano, podendo definir prioridades para cada uma delas. O resultado final será uma lista de ações elaboradas, ordenadas e prontas para serem executadas. O gerenciamento baseado em políticas (*Policy Based Network Management* (PBNM)) pode ser utilizado por todos os serviços do ciclo autonômico. No entanto, o PBNM é especialmente importante e freqüentemente aplicado pelo serviço de planejamento. Nesse serviço, o EA normalmente recorre a uma máquina de políticas disponível em sua base de conhecimento para definir o que deve ser feito de acordo com os resultados da análise disponíveis.

Gerenciamento baseado em políticas (PBNM): políticas podem ser definidas como um conjunto de considerações projetadas para guiar decisões. Elas definem objetivos e limites que governam as ações dos elementos autonômicos. O PBNM é uma abordagem utilizada para a automatização de tarefas ligadas à gerência de redes, que tem crescido e se popularizado nos últimos anos. A aplicação de políticas no contexto de redes com o objetivo de se obter gerenciamento, operação e controle do sistema automatizados caracteriza o PBNM. Políticas podem ser projetadas e aplicadas para grupos específicos de elementos de rede, recursos, serviços e usuários e podem estar relacionadas às diversas áreas funcionais (configuração, segurança, falhas, desempenho e contabilização) e níveis de gerenciamento (elemento de rede, gerencia de elemento de rede, gerencia de rede, gerencia de serviço e gerencia negócio) [Chadha et al., 2002]. As políticas podem ser, em geral, separadas em dois grandes grupos: políticas de obrigação e políticas de autorização. No primeiro grupo, as políticas são regras com um formato condição-ação, engatilhadas por um evento. Já no segundo grupo as políticas são usadas para se definir quais serviços ou recursos um elemento da rede (físico ou lógico) pode acessar [Damianou et al., 2002].

Para realização de um uso mais efetivo e abrangente desta abordagem, torna-se necessário a definição e padronização de arcabouços e modelos conceituais, funcionais e de informação. Vários grupos de pesquisa acadêmicos e empresariais têm trabalhado na construção de um

padrão que permita a melhoria da escalabilidade do gerenciamento de redes e facilidade na distribuição de políticas, inclusive em aplicações que utilizem redes compostas por equipamentos e soluções desenvolvidas por diferentes fabricantes. O uso do PBNM já possui um grande importância no gerenciamento de redes atualmente, principalmente em contextos de grandes escalas e diferentes níveis de complexidade e qualidade de serviço. Ele permite a automatização de tarefas trabalhosas e repetitivas. Com a evolução destes sistemas de gerenciamento, as políticas utilizadas se tornaram cada vez mais sutis e de alto-nível, independentes das tecnologias e algoritmos utilizados pela rede (políticas de negócio). A realização da tarefas passará então de automática para autônoma, e os administradores terão papel colaborativo na administração dos elementos da rede.

Um importante trabalho desenvolvido na área de uso de PBNM em sistemas autônomos foi publicado em [Agrawal et al., 2005]. Nele os autores definem uma plataforma *middleware* de políticas para computação autônoma, chamada *Policy Management for Autonomic Computing* (PMAC), capaz de gerenciar múltiplos aspectos de sistemas distribuídos de larga escala, tais como qualidade de serviço, configuração e auditoria. Os autores descrevem os modelos de informação e de sistema utilizados pelo PMAC para representação de políticas e interação entre componentes de políticas e recursos gerenciados.

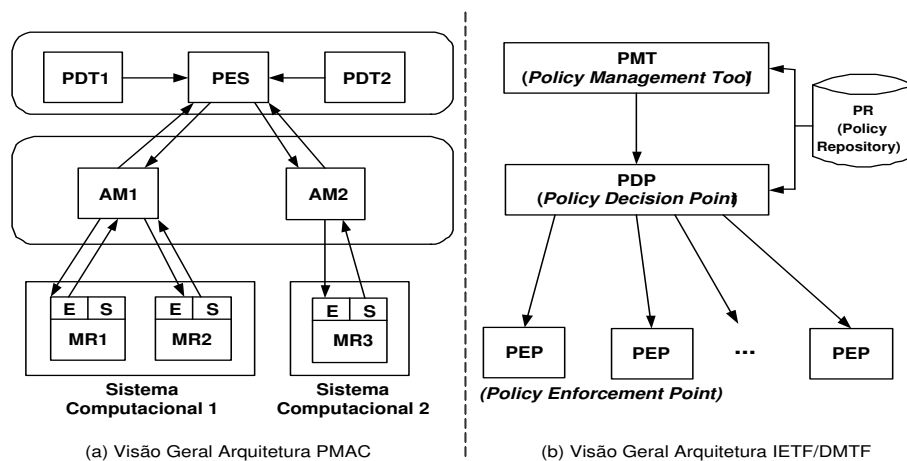


Figura 2.5: Arcabouço PMAC comparado a arquitetura IETF/DMTF tradicional.

O modelo de informação utilizado é baseado no modelo CIM (*Common Information Model*) [(DMTF), 1999] definido pela DMTF. Cada política é uma regra que contém quatro componentes: condições, ações, prioridades e papéis. Uma política é aplicável quando as condições associadas a ela são verdadeiras. Neste caso, o conjunto de ações associadas a esta mesma política são executadas. O valor da prioridade define qual política deve ser aplicada primeiramente quando existem múltiplas políticas aplicáveis com ações potencialmente conflitantes. Finalmente o papel define o contexto no qual a política será relevante. Em relação ao modelo de sistema, PMAC utiliza a arquitetura para computação autônoma baseada em EAs com gerentes autônomos (AMs) responsáveis por recursos gerenciáveis (MRs). Os AMs realizam um ciclo autônomo de monitoração, análise, planejamento e execução, utilizando para tal políticas definidas para os MRs. O relacionamento entre AMs e MRs é de 1:N, ou

seja, um AM é responsável por um ou mais MRs e cada MR é controlado por exatamente um AM.

Quatro componentes formam a estrutura de alto-nível do arcabouço PMAC: ferramenta de definição de políticas (PDT), o qual define uma interface gráfica de usuário que permite criação e modificação de políticas; Armazém de editor de políticas (PES) para deposição e persistência de políticas e meta-dados relacionados, agindo portanto como um repositório central utilizado por diversas ferramentas de definição de políticas; Um gerente autonômico (AM) e um ou mais recursos gerenciados (MRs). O AM é o componente central do PMAC, o qual é capaz de obter políticas do PES e registrar MRs que estejam interessados em receber orientação através de políticas. A Figura 2.5 ilustra a diferença entre o arcabouço proposto pelo PMAC e aquele definido na arquitetura de PBNM IETF/DMTF tradicional. O arcabouço de políticas proposto pela IETF/DMTF é composto por quatro elementos: ferramenta de gerenciamento (MT), repositório de políticas (PR), *policy decision point* (PDP), responsável por recuperar e interpretar as políticas armazenadas, e *policy enforcement point* (PEP), dispositivo capaz de aplicar e executar diferentes políticas, recebidas através de comunicação com PDP. Na comparação entre os dois modelos, o papel de um AM é similar ao de um PDP acrescido de outras funcionalidades tais como monitoração de estados, correlação de eventos e notificação. O MR por sua vez apresenta similaridades com o componente PEP tradicional. Os PDTs e PES da arquitetura PMAC possuem correspondência com PMT e PR do modelo IETF/DMTF.

2.5.3.4 Serviço de Execução

O serviço de execução está relacionado a um dos principais aspectos do paradigma da computação autonômica: a auto-configuração. É através deste aspecto que o EA consegue atuar sobre o hardware ou software do elemento gerenciado (definindo por exemplo ações de reconfiguração e reprogramação), buscando configurá-los da melhor maneira possível para as necessidades deste elemento e às condições do ambiente que o cercam. Este serviço não deve julgar se as ações executadas são adequadas ou não, pois a viabilidade e necessidade das mesma deve ter sido cuidadosamente avaliada na etapa de planejamento (Seção 4.4). Cabe a este serviço apenas executar o plano da forma como foi recebido e também tratar das eventuais falhas que possam ocorrer.

Reprogramação e Reconfiguração: existem dois importantes conceitos relacionados com a implementação da computação autonômica em sistemas de vários tipos: a reprogramação de software e reconfiguração de hardware. Dada a necessidade de se ajustar ao ambiente em que se encontra de uma forma contínua e dinâmica, um sistema autonômico deve possuir no mínimo soluções básicas destes dois conceitos. Através deles, será possível ajustar o software e/ou o hardware dos componentes do sistema, de forma a atender aos requisitos de controle e supervisão mais adequados ao momento. A reprogramação define a capacidade de se modificar parte ou mesmo todo o software utilizado, de forma a incluir novas funções necessárias no momento e eliminar aqueles que não são mais executadas. A reprogramação

pode ser realizada de várias maneiras, e uma delas pode ser com o uso de agentes móveis. Já a reconfiguração trata da utilização de hardware capaz de ser modificado também de forma a oferecer funcionalidades diferentes a cada momento, as quais podem ser utilizadas de forma independente e de acordo com a demanda [Hauser, 2000]. Existe disponíveis no mercado, hardware reconfigurável baseado na tecnologia de FPGA (*Field-Programmable Gate Array*). Outras abordagens mais elaboradas, porém com poucos resultados práticos atualmente, prevêem o uso de técnicas evolutivas em hardware reconfigurável. Nestes casos, o hardware é capaz de modificar sua estrutura interna, de forma a criar novas funcionalidades mais flexíveis e eficientes [Barlow e Edwards, 2001].

2.5.3.5 Base de Conhecimento (BC)

A Base de Conhecimento (BC) é o componentes da arquitetura de um EA responsável pelo armazenamento de dados, informações, conhecimento, políticas, algoritmos, dentre outros elementos. Todos os serviços do ciclo autônomo utilizam a BC, sendo que seus componentes devem ser portanto armazenados como conhecimento compartilhado. O conhecimento utilizado pelo gerente autônomo pode ser obtido de três maneiras: (i) o conhecimento é passado pelos elementos gerenciados para o gerente através de interfaces de atuação (*touchpoints*), (ii) o conhecimento é recuperado a partir de um serviço de informações externo e (iii) o próprio gerente autônomo produz conhecimento. Os tipos de conhecimento armazenados podem ser diversos: informações sobre componentes do(s) elemento(s) gerenciado(s), suas construções e configurações, conhecimento sobre configurações e instalações, métodos para definição de políticas que governam as tomadas de decisão dos gerentes autônomos, dados monitorados, sintomas, árvores de decisão e métodos para representação de conhecimento sobre determinação de problemas são alguns exemplos de conteúdos de uma BC. A determinação precisa dos elementos armazenados em um BC deve ser realizada de acordo com o ambiente, aplicação, componentes e objetivos da solução autônoma considerada [AC Blueprint, 2006].

Dentre os principais desafios relacionado com a construção de uma BC pode-se citar a escolha de métodos para representação de conhecimento [Sowa, 2000]. A representação do conhecimento consiste na utilização de métodos para se armazenar e manipular conhecimento em um sistema de informação de maneira formal, possibilitando que o mesmo seja utilizado em diversos mecanismos para realização de diferentes tarefas. Duas tecnologias importantes ligadas a desafios de representação e manipulação de conhecimento são o gerenciamento de conhecimento e as ontologias.

Gerenciamento de Conhecimento (*Knowledge Management (KM)*): é um conceito ligado a forma como elementos de um sistema adquirem, gerenciam e usam o conhecimento conquistado. Em um contexto organizacional, dados representam fatos ou valores de um resultado, relações entre dados representam informações, padrões das relações entre dados e informações representam conhecimento. O valor do gerenciamento de conhecimento se relaciona diretamente à eficácia com a qual essa técnica habilita EAs a lidarem com diversas situações e efetivamente visualizarem e criarem seu futuro. Sem a utilização do gerencia-

mento de conhecimento, cada situação é tratada com base no que os EAs trazem consigo para a mesma. No entanto, com a utilização dessa técnica, as situações são tratadas com a soma total de todo conhecimento adquirido até aquele ponto sobre o contexto atual ou similares [McElroy e W, 2002, Alavi e Leidner, 1999]. O gerenciamento de conhecimento deve ser realizado pelos serviços de análise e planejamento, e sua eficácia depende diretamente da armazenagem realizada pela BC.

Ontologias: as ontologias podem ser vistas como uma abordagem de modelagem conceitual expressiva e de alto-nível, para a descrição de conteúdos. As ontologias podem ser utilizadas como base para construção de algoritmos, ferramentas e modelos de correlação de dados e/ou eventos. Em essência, uma ontologia é uma teoria sobre a natureza da existência e os tipos de elementos que podem existir. No contexto da inteligência artificial, as ontologias descrevem uma conceitualização formal e compartilhada de um domínio de interesse em particular. Assim, as ontologias podem prover uma forma de se capturar um entendimento compartilhado de um domínio utilizado tanto por humanos quanto por sistemas para ajudar na troca e integração de informações. Em outras palavras, as ontologias definem uma representação explícita da semântica de dados, proporcionando interoperabilidade e integração de informações, e possibilitando o processamento dessas por máquinas e a construção de correlações com novo nível qualitativo de serviços [Stojanovic et al., 2004].

Outro conceito importante no contexto de EAs e que deve estar presente em todos os serviços apresentados nessa seção é o aprendizado de máquina.

Aprendizado de Máquina: todos os serviços que compõem o laço autônomo podem conter funções para aprendizado de máquina que permitam ao elemento autônomo aprimorar seus serviços e funções de gerenciamento. O aprendizado de máquina é uma sub-área da inteligência artificial que trata do desenvolvimento de algoritmos e técnicas para o aprendizado indutivo e dedutivo de sistemas computacionais. Existe uma grande sobreposição das áreas de aprendizado de máquinas e estatística, dado que ambas estudam a análise de um conjunto de dados. No entanto, considerando-se a área de aprendizado de máquina, existe uma grande preocupação quanto a complexidade computacional da implementação dos algoritmos, visto que grande parte deles não possui solução polinomial. Os algoritmos de aprendizado de máquina podem ser organizados de acordo com uma taxionomia baseada no desejo de retorno dos mesmos: aprendizado supervisionado, não-supervisionado, semi-supervisionado, reforço de aprendizado, dentre outros. Exemplos de técnicas e algoritmos para aprendizado de máquina são árvores de decisão, redes neurais artificiais, algoritmos genéticos e redes bayesianas [Weiss e Kulikowski, 1991, Mitchell, 1997].

Uma vez que os elementos autônimos em uma rede não existirão de forma isolada, e que ainda será necessário manter uma comunicação entre elementos de diferentes redes, é desejável que padrões de implementação para os mesmos, tanto para a estrutura de comportamento

interno quanto para o ciclo de vida, sejam adotado. É necessário também que um EA consiga realizar trocas com outros, e que todos os elementos desse tipo implementem uma solução que possa prover as características mínimas e os serviços básicos de uma rede autonômica. A seguir estão descritas as etapas que compõem os ciclos de vida e relacionamento de EAs. Vale ressaltar que a definição desses ciclos para as RSSFs está fora do escopo desse trabalho.

2.5.4 Ciclo de Vida de um EA

Uma série de etapas devem ser seguidas pelo próprio EA, e em alguns momentos pelo administrador da rede, para se criar, embutir e atualizar estes elementos. Essas tarefas podem ser executadas em momentos distintos, tais como antes, durante e após a criação do EA e em conjunto definem um ciclo de vida, percorrido periodicamente, até que se atinja uma situação de esgotamento, deficiência ou inadequação, e neste caso, a tarefa final de desinstalação e configuração é acionada [Kephart e Chess, 2003].

1. **Projeto, Teste e Verificação:** o projeto de um EA consiste na representação de necessidades, preferências e capacidades. As tarefas de realização de testes e verificações de EAs, ou seja, criação de mecanismos que permitam observar se determinado projeto de EA apresenta comportamento correto, são difíceis e delicadas.
2. **Instalação e Configuração:** o EA deverá basicamente seguir um processo de amarração (*bootstrapping*), o qual inclui a realização de um registro próprio em um diretório de serviços, publicando sua capacidade e informações de contato. Todos os EAs da rede podem utilizar o diretório para descobrir potenciais fornecedores e consumidores de informações e serviços.
3. **Monitoração e Determinação de Problemas:** os EAs devem monitorar a si próprios, seus fornecedores e consumidores de serviços e recursos, para garantir que estão cumprindo seus objetivos e armazenar essa informação para servir como base para adaptação, auto-otimização e reconfiguração por exemplo.
4. **Atualização:** será necessário que o EA se atualize em intervalos regulares de tempo. A determinação dos momentos em que são necessárias ações de atualização não é um problema trivial. Quando o EA define que está ultrapassado, precisando de novas funcionalidades e serviços, ele deve encontrar uma maneira de identificar que atualizações estão disponíveis, buscá-las e incorporá-las a si próprios.
5. **Desinstalação e Reposição:** eventualmente, após a realização de um grande número de atividades e instalação do maior número de atualizações possíveis, um EA deve identificar a necessidade de deixar a rede, ser desinstalado e substituído. Neste caso ele deve retirar do diretório de serviços suas ofertas de recursos e serviços, finalizar todos os acordos firmados com demais EAs daquela rede, anunciar ao administrador de rede a necessidade de um novo EA (se for o caso) ou mesmo instalá-lo, e finalmente executar funções de auto-eliminação.

2.5.5 Ciclo de Relacionamento entre EAs

Normalmente, em qualquer situação e tipo de sistema em que esteja instalado, um EA não existirá isoladamente. Em um contexto de redes autonômicas, os EAs de cada elemento de rede devem se relacionar uns com os outros de modo a controlar principalmente os enlaces que os ligam. Além disso, a interação entre EAs é fundamental para realização de compartilhamento de recursos e serviços necessários para se manter a produtividade e qualidade dos serviços de auto-gerenciamento providos [Kephart e Chess, 2003].

1. **Especificação:** nesta etapa, cada EA deve associar a si um conjunto de serviços ofertados e de serviços requisitados, ambos expressos em um formato padrão, para que os demais EAs da rede possam entendê-los.
2. **Localização:** uma vez publicados os serviços ofertados e requisitados, um EA deve ser capaz de localizar os serviços e recursos de que necessita, assim como ser encontrado pelos EAs que precisam dos serviços e recursos que oferece. Os elementos devem contatar potenciais provedores diretamente, determinando as regras do relacionamento.
3. **Negociação:** o processo através do qual EAs tentarão entrar em algum acordo sobre o relacionamento é chamado de negociação. Nele os elementos definirão as regras de provisionamento/obtenção de serviços ou recursos. Um EA somente presta serviços/recursos para outro, caso ele próprio não necessite dos mesmo, e julgue a parceria com outros EAs será interessante, justa e compensadora.
4. **Provisionamento:** uma vez firmado o acordo de negociação, os EAs devem então se comprometer a prover serviços ou recursos uns para os outros de acordo com as regras determinadas. Cada EA deve de alguma maneira registrar os compromissos firmados com outros EAs.
5. **Operação:** no momento em que um EA está efetivamente provendo serviço ou recurso para outro, o relacionamento entre eles está em operação. Durante este período, os gerentes instalados no provedor e no receptor devem monitorar toda a operação para garantir que todas as regras previamente determinadas estão sendo cumpridas por ambas as partes. No caso de violação de algum acordo, os EAs devem buscar uma solução apropriada.
6. **Terminação:** uma vez terminado o período do relacionamento, as partes concordam em terminá-lo, liberando seus recursos internos para outros fins e finalizando acordos para recursos ofertados que não mais são necessários.

2.6 Construção de uma Rede Autonômica

Nesse trabalho propomos uma estratégia para construção de uma rede autonômica composta por três passos distintos. Cada um destes passos trata de aspectos importantes no desenvolvimento dessas redes e está ligado a um ou mais conceitos discutidos nesta seção. A realização

dos três passos permite que uma solução autonômica completa seja desenvolvida e utilizada. O nível de autonomicidade da solução como um todo depende diretamente das propostas e algoritmos construídos e utilizados em cada um dos passos. A seguir estão descritos os três passos para a definição de uma solução autonômica:

1. **Escolha ou definição de uma arquitetura de gerenciamento:** a estratégia de controle e supervisão a ser seguida pela rede autonômica deve ser inicialmente definida por uma arquitetura de gerenciamento, a qual deve ser composta pelos modelos de informação, funcional e físico, e também pelos serviços e funções de gerenciamento considerados. A arquitetura deve introduzir uma organização para o gerenciamento, considerando as áreas funcionais e níveis de gerenciamento adequados, além de quaisquer outras perspectivas e informações relevantes. De acordo com o tipo de rede gerenciada, será mais adequado escolher dentre arquiteturas já existentes na literatura ou propor um novo arcabouço, capaz de se adequar melhor às particularidades dos elementos gerenciados e as ligações entre eles. É interessante que a arquitetura utilizada já esteja definida de acordo com princípios de auto-gerenciamento.

Para a solução de RSSFs autonômicas apresentada neste trabalho, a arquitetura de gerenciamento Manna foi escolhida. Esta arquitetura já prevê o uso de aspectos autonômicos no gerenciamento dos nós sensores, sendo portanto bastante adequada para utilização em uma solução geral de controle autonômico. Assim, todos os aspectos dessa solução implementados em última instância pelo elemento autonômico são referentes as diretivas de gerenciamento propostas pela arquitetura Manna. Mais detalhes sobre a arquitetura Manna podem ser encontrados na Seção 3 e em [Ruiz, 2003].

2. **Projeto do elemento autonômico:** uma vez definida a arquitetura, será necessário modelar o comportamento (ciclo autonômico) do elemento autonômico que deverá ser instalado em cada um dos elementos da rede. Conforme mencionado anteriormente, o menor nível de granularidade considerado em uma rede autonômica é o próprio elemento de rede e não seus componentes, os quais podem (e até devem) ser formados por hardware e software autonômico. Assim, o projeto de elemento autonômico deve ser realizado para estes elementos, independente da forma como seus componentes internos são gerenciados. O EA é composto pelo gerente autonômico e pelo(s) objeto(s) gerenciável(eis), que neste caso são os dispositivos computacionais que compõem a rede (computadores pessoais, laptops, PDAs, nós sensores, dentre outros). O ciclo autonômico executado pelo gerente deve seguir as diretrizes definidas pela arquitetura de gerenciamento utilizada, os limites impostos pelas características da rede e os objetivos da aplicação em particular. Assim, é possível se propor um modelo de EA genérico, e a partir dele, criar instâncias para o tipo de rede autonômica, e uma nova instância para a aplicação específica. Também pode ser necessária a definição dos *touchpoints* para os elementos gerenciáveis envolvidos, permitindo contato entre gerente autonômico e os mesmos, com menor complexidade.

O segundo passo no desenvolvimento de uma RSSF autonômica envolve a elaboração de um modelo para um EA, chamado Elemento Sensor Autonômico (ESA), que possa ser embutido nos componentes desses tipos de rede. Esse EA deve ser desenvolvido de forma flexível, possibilitando que as RSSFs sejam tão autonômicas quanto possível, de acordo com a disponibilidade dos recursos e os objetivos do gerenciamento. Além disso, esse EA deve ser instanciável para diferentes plataformas de nós sensores e tipos ou configurações de RSSFs. Uma vez que a construção de *touchpoints* para interfaces gerenciáveis dos recursos gerenciáveis de nós sensores não está no escopo desse trabalho, assumimos que tais interfaces já se encontram implementadas nesses elementos. A construção de um modelo para o ESA é a principal contribuição desse trabalho, e é apresentada em detalhes no Capítulo 4 a seguir.

3. Definição do ciclo de vida e de relacionamento dos elementos autonômicos:

a realização dos dois passos anteriores já permite que os objetivos e possibilidades de controle e supervisão estejam claros, e que a forma como os mesmos serão executados esteja modelada em um projeto de elemento autonômico. Somente com estes dois passos, uma solução autonômica está parcialmente definida e a rede poderia ser criada, instalada e iniciar suas operações autonômicas. No entanto, os EAs criados executariam seus ciclos autonômicos sem qualquer controle e não poderiam se comunicar uns com os outros para realizar a troca de recursos e serviços. Para tanto é necessária a criação dos modelos para o ciclo de vida e de relacionamento dos EAs. Com eles, cada EA pode gerenciar seu funcionamento, desde a etapa de instalação até a desinstalação e substituição, e também controlar a negociação para realização de trocas com os outros EAs da mesma e de outras redes. Desta forma, os administradores humanos ficam definitivamente fora do círculo de controle, e assumem o papel de supervisores. Não seria mais necessária a intervenção humana direta para que os elementos da rede sejam gerenciados e possuam um relacionamento entre si.

Após a definição da arquitetura de gerenciamento utilizada (arquitetura Manna) e da construção de um modelo para o elemento autonômico capaz de controlar os componentes da rede (ESA), a última tarefa para o projeto completo de uma RSSF autonômica, é a modelagem dos ciclos de vida e de relacionamento dos ESAs. Essa definição também é um desafio de pesquisa, e assim como os demais passos descritos acima, deve ser desenvolvida observando-se as características particulares da rede em questão, e considerando a implementação de padrões consolidados e abertos. Até o momento do desenvolvimento deste trabalho, não foram encontrados na literatura publicações relacionadas com a definição dos ciclos de vida e relacionamento dos ESAs. O desenvolvimento desse passo como um todo está fora do escopo dessa dissertação. Em nossos experimentos, avaliamos uma RSSF cujos ESAs realizam troca de informações, mas não negociam recursos ou serviços. Assim, não foi necessária a definição e implementação do ciclo de relacionamento entre ESAs e das interfaces de comunicação (*touchpoints*) entre eles.

2.7 Considerações Finais

Este capítulo apresentou conceitos importantes relacionados com o paradigma da computação autonômica aplicado ao contexto das redes. Foram apresentados os níveis de autonomicidade que definem o quão autonômico é um sistema ou rede, ou seja, qual o papel dos gerentes humanos e a importância deles para na execução de tarefas de gerenciamento. Os principais serviços autonômicos também foram discutidos, considerando-se a aplicação dos mesmos em diferentes ambientes de redes, e em particular em RSSFs. Em seguida, a menor parte de uma rede autonômica, o elemento autonômico, foi introduzido. Os componentes dos EAs (gerente autonômico e elemento gerenciado), bem como o funcionamento do laço autonômico de controle infinito e os serviços que compõem o mesmo foram explicados. Algumas das tecnologias relacionadas com cada um desses serviços foram brevemente explicadas. Uma vez que essa dissertação propõe um EA específico para RSSFs, é importante que se defina o que são esses elementos, seus componentes e seu comportamento, assim como a importância que os mesmos possuem na definição de uma solução autonômica completa. Os ciclos de vida e de relacionamento entre EAs também foram apresentados nesse capítulo, mas a definição dos mesmos para RSSFs está fora do escopo desse trabalho. Finalmente, um conjunto de passos propostos como uma possível metodologia para o desenvolvimento de redes autonômicas foi descrito, demonstrando-se qual o posicionamento dos objetivos desse trabalho no escopo dessa metodologia. O capítulo a seguir descreve os trabalhos relacionados com computação autonômica, e em particular redes autonômicas publicados na literatura, indicando as contribuições específicas de cada um deles e comentando as diferenças de objetivos dos mesmos em relação ao trabalho apresentado nessa dissertação.

Capítulo 3

Trabalhos Relacionados

Esse capítulo apresenta os artigos encontrados na literatura cujos temas e contribuições foram consideradas relacionados àqueles apresentados por esse trabalho. Inicialmente, na Seção 3.1, descreveremos algumas das ferramentas autonômicas desenvolvidas por empresas de TI para sistemas tradicionais. Em seguida, as soluções autonômicas encontradas em artigos direcionados para redes tradicionais com fio são apresentados e discutidos na Seção 3.2. Os trabalhos que abordam aspectos autonômicos específicos para redes sem fio, e em particular RSSFs, são apresentados pela Seção 3.3. A arquitetura de gerenciamento utilizada como base de desenvolvimento para o elemento sensor autonômico proposto nesse trabalho é discutida e apresentada em detalhes na Seção 3.4, e as principais tecnologias para o gerenciamento tradicional de redes de computadores na Seção 3.5. O capítulo apresenta o panorama do desenvolvimento de trabalhos acadêmicos e da indústria, sobre aplicação de aspectos autonômicos em diferentes contextos, disponíveis atualmente na literatura. A Seção 3.6 apresenta um resumo sobre o que foi apresentado nesse capítulo.

3.1 Computação Autonômica para Sistemas Tradicionais

Em termos de soluções autonômicas para sistemas tradicionais, a IBM tem lançado nos últimos anos várias ferramentas para sistemas de TI em seu projeto *AlphaWorks Autonomic Zone* [Alphaworks, 2006], tais como ferramentas de análise que correlacionam logs de eventos para identificação de padrões, a ferramenta *Tivoli Autonomic Monitoring Engine* que ajuda a realizar a análise de causa de problemas e ações corretivas automáticas e a ferramenta ABLE, que é usada para análises mais complexas utilizando técnicas de aprendizado, séries temporais, classificação bayesiana, entre outras.

A IBM também possui um conjunto de ferramentas, componentes, cenários e documentações com tecnologia autonômica para auto-gerenciamento, chamado *Autonomic Computing Toolkit (ACT)* [Autonomic Toolkit, 2006]. Esse conjunto foi projetado para usuários de sistemas tradicionais que desejam aprender, adaptar e desenvolver comportamento autonômico em seus produtos e sistemas. Os componentes do ACT listados acima podem ser divididos em três categorias: (1) determinação de problemas: representa um sistema simples de auto-

gerenciamento que usa um laço de controle inteligente para coletar informações, analisá-las, planejar respostas apropriadas e então realizar os ajustes necessários para resolver problemas; (2) deposição e instalação de soluções: permite que o software seja depositado e controlado com reduzida interação humana, e (3) console de soluções integradas: utilizado para se construir capacidades comuns para administração de sistemas. Provê a estrutura para ajudar a consolidar a interface de usuário em um ambiente de TI heterogêneo.

Uma outra solução autônômica projetada pela IBM chamada AIDE (*Autonomic Integrated Development Environment*) [AIDE, 2006] é um conjunto de ferramentas para desenvolvimento de software que contempla padrões para gerenciamento distribuído de *Web Services* (WSDM - *Web Services Distributed Management*), capaz de facilitar os problemas de gerenciamento de TI por meio da definição de uma arquitetura para conexão de recursos computacionais e aplicações de gerenciamento com interfaces baseadas em padrões. Um gerente autônômico se comunica com um ou mais recursos (bancos de dados, servidores, dentre outros) através de *touchpoints* [AC Blueprint, 2006]. Os *touchpoints* podem ser vistos como uma estreita camada de código que realiza tradução entre padrões de interface e comandos específicos de produtos. Além das soluções, software e ferramentas descritas acima, a IBM possui outras opções autônômicas adequadas para diferentes necessidades e contextos. Em [Alphaworks, 2006] podem ser encontradas descrições, tutoriais e em alguns casos até mesmo versões para teste dessas opções.

Outras empresas do setor de TI também estão envolvidas com o paradigma da computação autônômica, anunciando planos e projetos nessa área. A Microsoft possui um projeto para computação autônômica chamado *Dynamic Systems Initiative* (DSI), o qual deve se concentrar no tratamento eficiente de demandas flutuantes por recursos de sistema [DSI, 2006]. O DSI é um comprometimento entre a Microsoft e suas empresas parceiras para ajudar os times de TI a capturar e utilizar conhecimento para projetar sistemas mais fáceis de se gerenciar e automatizar operações em execução, reduzindo custos e liberando esses times para que os mesmos possam manter o foco, de uma forma pró-ativa, no que é verdadeiramente importante para seus negócios. O desenvolvimento do DSI envolve a criação de soluções tais como ferramentas de desenvolvimento, sistemas operacionais, ferramentas de virtualização, ferramentas de gerenciamento, e aplicações. Várias empresas, tais como Dell, Intel e Sun, são parceiras da Microsoft nesse projeto e também estão projetando soluções que permitam que a visão de auto-gerenciamento do DSI englobe ambientes com hardware e software heterogêneo.

A Sun Microsystems possui uma iniciativa ligada a computação autônômica chamada *N1 Plan* [N1 Plan, 2006]. Um software desenvolvido para essa iniciativa, chamado *N1 Software*, é um portfólio de produtos para automação do ciclo de vida do gerenciamento de servidores e aplicações, e gerenciamento de serviços de grades computacionais em ambientes heterogêneos. Os principais problemas de negócios atendidos pelos produtos desse portfólio são: (1) aumento da velocidade de desenvolvimento de novos serviços, automatizando software complexo de instalação e configuração em infra-estruturas de redes heterogêneas, e ajudando a acelerar o serviço de deposição; (2) diminuição de custos operacionais, habilitando compartilhamento de recursos de sistemas para permitir maior flexibilidade para alocação dos mesmos quando

houverem demandas; (3) melhoria da eficiência de negócio, provendo aumento consistente das capacidades e eliminando erros de configuração.

A Hewlett-Packard (HP) também já lançou sua própria infra-estrutura autonômica chamada *Utility Data Center* (UDC) [UDC, 2006]. O desafio enfrentado pela empresa nesse caso é o de entender quais iniciativas de gerenciamento de TI podem prover o melhor retorno, dividindo-a em passos gerenciáveis e criando um guia realístico para execução dos mesmos. A indústria de TI passa por uma transformação fundamental de mudança do foco de investimento, da manutenção para a inovação de negócios. A HP desenvolveu um modelo que facilita essa transformação da indústria de TI, simplificando a forma como os processos podem ser separados, melhorando capacidades de gerenciamento de software e alinhando-as melhor com as prioridades de negócio. Esse modelo ajuda empresas a avaliar seus estados atuais de maturidade em TI, determinando as áreas apropriadas para se introduzir iniciativas para transformações de TI. Esse modelo é formado por três camadas: (1) otimização de infra-estrutura, que define o uso de uma arquitetura tecnológica de gerenciamento heterogênea e integrada, permitindo um ambiente consolidado que facilite aplicações de gerenciamento fim-a-fim; (2) automação dos processos de TI, prevê que as empresas de TI adaptem, apliquem e automatizem processos de TI padrão e melhores práticas baseadas em *IT Infrastructure Library* (ITIL); (3) alinhamento de negócios, que alinha suporte e entrega de serviços com requisitos de negócio e demandas de consumidores. A chave para essa camada é relacionar serviços de gerenciamento de TI com processos específicos de negócio, indicadores chave de desempenho e políticas para gerenciamento de riscos, de uma forma amplamente automatizada.

A NEC é uma empresa desenvolvedora de soluções de TI que trabalha em particular na fabricação e implementação de hardware e software ubíquo e que também possui sua plataforma para computação autonômica. O VALUMO é uma plataforma composta por um grupo de tecnologias desenvolvidas pela NEC para facilitar operação e integração de sistemas, melhorando aspectos tais como confiabilidade, disponibilidade e operabilidade. Uma das características dessa plataforma é o provisionamento de autonomia para os sistemas, que devem ser capazes de contribuir para estabilização das operações, monitorando seu próprio status, analisando causas de problemas e executando operações automáticas de acordo com políticas, depois que as causas forem identificadas [VALUMO, 2006].

3.2 Computação Autonômica Aplicada a Redes Com Fio

Em relação à utilização do paradigma de redes autonômicas no desenvolvimento de soluções para redes tradicionais, existem alguns trabalhos publicados na literatura que tratam do desenvolvimento de técnicas e algoritmos para implementação de serviços específicos, tais como auto-organização, auto-configuração e reforço de aprendizado, por exemplo. Todas essas abordagens no entanto, foram definidas e projetadas para redes tradicionais, podendo não ser adequadas ou mesmo diretamente aplicáveis em redes sem fio e em particular em RSSFs, uma vez que não consideram as características intrínsecas desses tipos de rede.

[Littman et al., 2004] descrevem a implementação de um sistema capaz de aprender como restaurar a conectividade de uma rede de forma eficiente, após a ocorrência de falhas. Esse trabalho, segundo os autores, complementam pesquisas anteriores sobre estudo e classificação sistemática de falhas, provendo um arcabouço para introdução de uma forma flexível de obtenção de informações e ações de reparo ou cura, sem a necessidade de ajuste manual detalhado. O trabalho explora a técnica de reforço de aprendizado chamada *Cost-Sensitive Fault Remediation (CSFR)*. Uma árvore de políticas para execução de testes e ações é montada, através do uso de programação dinâmica. Essa árvore indica quais testes e ações devem ser executadas em quais situações, inclusive definindo a ordem de precedência entre os mesmos. O artigo apresenta resultados de experimentos que mostram como a solução proposta minimiza o tempo gasto por uma rede para execução de ações de auto-cura.

Em [Eymann et al., 2003], os autores apresentam a avaliação de um mecanismo descentralizado para alocação de recursos em redes da camada de aplicação (*Application-layer Networks - ALN*), tais como redes P2P e Grids, com base em um modelo econômico. O modelo utilizado é distribuído e prevê que cada elemento do sistema deve buscar a maximização de seus interesses próprios. Agentes de software realizam cooperação para compra e venda de recursos e serviços entre si. Em resumo, os elementos da rede se auto-organizam em termos de alocação de recursos e serviços (processamento, espaço em disco, dentre outros) de forma descentralizada e considerando a busca pelo benefício próprio e o caráter limitado desses recursos. Simulações foram realizadas para comparação da eficiência da auto-organização de recursos, obtida com um modelo econômico centralizado e o modelo distribuído adotado. Diferentes densidades de rede e comportamentos dos elementos da rede (fluxo de entrada e saída de nós) foram testados. Os resultados mostraram que o modelo descentralizado possui melhores resultados para a variação de densidade e praticamente não é alterado na variação de dinamicidade da rede.

Um estudo sobre a implementação de auto-configuração de serviços, ou seja, capacidade de um sistema de configurar seus próprios serviços de rede e aplicações em resposta às necessidades dos usuários e características do ambiente, para redes autonômicas é apresentado por [Melcher e Mitchell, 2004]. Esse tipo de serviço autonômico é importante para as redes desse tipo, uma vez que leva à redução de complexidades administrativas, erros causados por operadores humanos, dentre outros aspectos. Os autores apresentam várias considerações sobre tipos de ambientes ou redes que necessitam da auto-configuração, tais como redes de sistemas complexos de TI, redes móveis sem fios e redes locais residenciais com e sem acesso à Internet. Em seguida, requisitos de implementação para uma solução autonômica de auto-organização aplicável em todos os contextos considerados são apresentados. Uma avaliação sobre as tecnologias atuais de configuração de serviços de rede (DNS, NIS, DHCP, dentre outros) é realizada no intuito de se verificar a aplicabilidade dos mesmos em uma solução autonômica. Os autores concluem que tais solução não são aplicáveis uma vez que nenhuma delas (isoladamente ou em grupo) atendem aos requisitos propostos. Finalmente, adições e modificações são propostas para as tecnologias existentes, de forma a tratar os inconvenientes apontados e prover amplamente serviços de redes autonômicas para vários tipos de ambien-

tes. Além disso, o trabalho descreve como utilizar o IBM *Autonomic Computing Toolkit* para desenvolver a solução de auto-organização de serviços.

Além dos artigos citados acima, existem ainda alguns outros trabalhos que tratam de aspectos específicos do paradigma da computação autônoma para redes tradicionais, tais como em [Norman et al., 2003], [Appavoo et al., 2003], [Russel et al., 2003], [Chess et al., 2003] e [Lanfranchi et al., 2003].

Também foram encontrados na literatura, dois trabalhos que tratam da definição de um elemento autônomo para ambientes de redes tradicionais particulares. Esses trabalhos, apesar de possuírem objetivos bastante semelhantes aos dessa dissertação, não apresentam propostas diretamente aplicáveis no contexto de RSSFs. Ao contrário, os EAs apresentados são específicos e possuem implementações internas próprias para os ambientes para os quais foram projetados.

Em [Tian et al., 2005], os autores propõem uma arquitetura para um ambiente *web service* autônomo, baseado em técnicas de programação reflexiva, ou seja, capaz de inspecionar e adaptar seu comportamento interno em resposta à mudanças de condições. Os componentes do site hospedeiro do sistema *web service* se ajustam e colaboram para prover um sistema auto-gerenciado e auto-otimizado. Esses sites hospedeiros são compostos por dois níveis de gerenciamento: no nível de componente, cada elemento de um sistema *web service* é responsável por gerenciar seu próprio desempenho e objetivos especificados pelo gerente do site. Para tanto, um elemento autônomo é responsável por cada componente. No nível do site hospedeiro, um gerente monitora o cumprimento das diretivas de acordos de serviços (SLAs), asseguradas ao cliente do sistema. [White et al., 2005] descrevem um projeto em hardware de um elemento autônomo, onde um cartão é inserido em um slot PCI e controla um servidor, através de comunicação PCI, IPMI e interfaces de rede. Segundo os autores, os elementos autônomos devem ser projetados em um plano separado, ou seja, em uma plataforma com recursos computacionais e de armazenamento separados, uma vez que consideram essencial que o gerente autônomo seja computacionalmente distinto da entidade que gerencia. São providas descrições de alguns cenários de gerenciamento, da arquitetura da máquina autônoma e de como o ciclo de vida das tomadas de decisões por delegação deve ser gerenciado.

3.3 Computação Autônoma Aplicada a Redes Sem Fio e RSSFs

Com relação às redes sem fio, existem poucos trabalhos disponíveis na literatura que propõem soluções para auto-gerenciamento desse tipo de rede. Em [Velayos e Karlsson, 2005] são apresentadas extensões para a arquitetura de redes locais sem fios para dar suporte à deposição interativa de conectividade nessas redes. O trabalho apresenta investigações ligadas a auto-configuração de redes locais sem fios em três níveis. No nível de rede, é estudada como um conjunto de pontos de acesso conectados a diferentes redes IP podem colaborar para a construção de uma única rede sem fio local. Em um nível inter-celular, sendo a célula a área de abrangência de um ponto de acesso, o trabalho investiga como pontos de acessos vizinhos

podem cooperar para garantir *handoffs* ou trocas de células, com bom desempenho e rapidez. No nível intra-celular, os autores estudam mecanismos de controle de admissão para proteger as células contra congestionamentos.

O trabalho descrito em [Zimmermann et al., 2005] apresenta uma abordagem descentralizada para o gerenciamento autônomo de um grupo de estações base que colaboram entre si, para prover acesso a redes sem fio eficiente e efetivo em ambientes dinâmicos. Uma plataforma que dá suporte para várias funções de gerenciamento diferentes baseada em mecanismos comuns de troca de informações é apresentada. Uma característica central do sistema é a inclusão de um mecanismo de realimentação de informações monitoradas no processo de gerenciamento autônomo, o qual pode melhorar a operação do sistema de gerenciamento e a qualidade de suas decisões. A avaliação preliminar da implementação de um protótipo mostra que o sistema de gerenciamento autônomo é escalável. Outros experimentos demonstram a possibilidade de se detectar problemas da rede causados por interferência de rádios ou ataques ativos de uma forma autônoma.

Existem alguns trabalhos disponíveis na literatura que tratam do uso de serviços ou aspectos autônomos em RSSFs. Todos esses trabalhos demonstram os benefícios da aplicação de alguma característica particular do paradigma de redes autônomas. No entanto, nenhum deles apresenta um modelo para tornar o elemento de rede das RSSFs (nó sensor) autônomo. Em [Marsh et al., 2004] os autores discutem como características autônomas podem ser incorporadas em RSSFs utilizando a deposição de tecnologias móveis e ágeis baseadas no uso de múltiplos agentes. Inicialmente conceitos de RSSFs, computação autônoma e o uso de agentes neste paradigma são discutidos. Em seguida foram definidos o cenário considerado como estudo de caso e os nós sensores utilizados (Mica Motes). Os autores definem o cenário de simulação como um sistema nervoso ambiental (*Environmental Nervous System* - ENS) que pode prover detecção e resposta automática de eventos do ambiente. Uma RSSF contendo três nós sensores foi utilizada no experimento realizado neste trabalho. Cada nó dessa rede foi programado de forma diferente, e os resultados mostraram que o nó autônomo foi capaz de detectar eventos com mais eficiência além de enviar um número maior de dados para o PA. A relação para o número de transmissões entre os nós autônomo e o tradicional foi de 2.5:1. A capacidade de detecção do nó autônomo foi de 91% contra 77% do nó tradicional.

No artigo descrito em [Pujolle e Chaouchi, 2005], os autores apresentam uma arquitetura autônoma (*Autonomic-Oriented Architecture* - AoA) definida para dar suporte de auto-organização e auto-gerenciamento em redes de sensores. A arquitetura AoA é definida como um conjunto de 4 planos de abstrações, sendo eles os planos de gerenciamento, conhecimento, controle e dados. Não é o objetivo do artigo propor novos algoritmos, protocolos ou esquemas para o plano de controle, mas sim prover uma forma de se realizar seleção dinâmica dos melhores algoritmos, protocolos e valores de parâmetros a qualquer momento. Este é um modelo de comunicação inteligente que usa diferentes protocolos de transporte e rede dependendo do estado e objetivo corrente da rede. Para implementar o mecanismo inteligente de escolha da melhor pilha de protocolos, é proposto o uso de agentes. Cada sensor possuirá dois agentes associados: um de baixo nível, simples e reativo e outro de alto nível, que faça parte de um

sistema multi-agente inteligente distribuído sobre a rede de sensores. O agente inteligente é capaz de adquirir e processar informações, permitindo que o agente de baixo nível se adapte as diferentes situações.

Outros artigos disponíveis na literatura também utilizam aspectos de computação autônoma para implementação de algum serviço ou função, como auto-organização, auto-manutenção e auto-configuração em RSSFs. Este trabalho difere dos artigos citados acima por não apresentar soluções autônomas para serviços específicos, e sim o projeto de um ESA capaz de possibilitar a implementação integrada de serviços do paradigma de redes autônomas nos componentes das RSSFs. Em [Silva et al., 2004b] os autores apresentam uma solução baseada em políticas para auto-configuração e auto-organização de RSSFs, formando grupos de nós e controlando a densidade por meio da indicação de redundâncias, utilizando uma estratégia de gerenciamento distribuída. Foram realizados experimentos para avaliar a aplicabilidade e desempenho da solução para formação de grupos e balanceamento de RSSFs heterogêneas. Três tipos de cenários foram considerados: cenário com todos os nós sensores posicionados em uma grade, cenário com todos os nós sensores distribuídos aleatoriamente e cenário com nós mais poderosos posicionados em uma grade e nós comuns distribuídos aleatoriamente. Em todos os casos, as RSSFs possuíam 196 nós comuns e 16 nós com maior capacidade de hardware, comunicação e energia. Os resultados mostraram que, para todos os cenários, os objetivos foram alcançados. O impacto do consumo de energia com comunicação e processamento apresentou pouca relevância, dadas as vantagens e benefícios oferecidos pela solução.

O trabalho descrito em [Liu e Martonosi, 2003] propõe um *middleware* e uma API para adaptatividade e atualização (auto-configuração) de aplicações em RSSFs. Novos protocolos podem ser adicionados e mudanças entre protocolos podem ser executadas. Assim, quando determinado algoritmo, protocolo ou aplicação não estiver apresentando desempenho satisfatório, o elemento de rede pode decidir trocá-lo a fim de adaptar-se às condições próprias e/ou do ambiente onde se encontra. Os autores propõem um modelo baseado em camadas que permite tal adaptabilidade. No entanto, esse modelo contempla apenas mudança entre diferentes implementações para algoritmos de roteamento. Segundo os autores, o uso desse sistema permite aumento de confiabilidade, desempenho e eficiência no consumo de energia. Para avaliação, foram realizados experimentos utilizando-se dois *handhelds* iPAQ e duas opções de algoritmos de roteamento. Além disso, simulações foram executadas por meio de uma ferramenta chamada ZnetSim, contemplando cinquenta nós e três opções de algoritmos para roteamento. Os parâmetros observados em ambos os casos foram consumo de energia, desempenho do roteamento e da reprogramação, tolerância a falhas e *overhead*.

Em [Figueiredo et al., 2005], os autores discutem o uso de políticas, uma abordagem bastante conhecida no gerenciamento de redes de computadores, para o estabelecimento de interações locais entre os elementos de uma RSSF. Essa abordagem define uma maneira flexível e de alto-nível para realização de tarefas de gerenciamento. Em particular, o uso de políticas é aplicado na tarefa fundamental de roteamento em RSSFs auto-organizáveis. Os nós sensores de uma RSSF hierárquica são programados com uma política que permite aos mesmos

escolher entre duas opções de algoritmos de roteamento: reativa e pró-ativa. Em intervalos regulares de tempo, os nós da rede avaliam essa política. Se existirem muitos nós enviando dados para o líder do grupo, um algoritmo de roteamento pró-ativo é utilizado. Caso contrário, o algoritmo reativo é preferido. Experimentos foram realizados com uso da ferramenta de simulação *Network Simulator* (NS) [Network Simulator, 2006]. Os resultados de simulação revelam que o uso de políticas na auto-organização de redes de sensores pode prover economia de recursos e benefícios.

Em [Machado et al., 2005] é apresentado um algoritmo de disseminação de dados para RSSFs, que combina conceitos apresentados por um algoritmo de encaminhamento baseado em trajetória com as informações providas pelo mapa de energia da rede, para determinação de rotas de uma forma dinâmica, de acordo com o nível de energia apresentado pelos nós sensores. Segundo os autores, esse algoritmo provê uma importante característica de um sistema autônomo, que é a adaptação de comportamento de acordo com recursos disponíveis. Simulações foram realizadas para se verificar o desempenho do algoritmo proposto. Em [Assunção e Ruiz, 2006] os autores propõe o uso do ITIL (*IT Infrastructure Library*) [ITIL, 2006] em conjunto com o paradigma da computação autônoma no projeto de uma RSSF capaz de realizar auto-cura. A RSSF auto-gerenciada proposta é capaz de se adaptar dinamicamente com o objetivo de manter a disponibilidade de serviços e promover a produtividade de seus recursos, por meio da implementação de técnicas para descoberta, exame, diagnóstico e reação a disfunções. De acordo com os autores, os resultados obtidos mostraram que a solução de auto-cura implementada foi capaz de estender a longevidade e disponibilidade da RSSF.

Existem diversos outros trabalhos publicados na literatura que tratam do desenvolvimento e implementação isolada de serviços autônomos específicos, dentre eles auto-organização, auto-configuração e auto-otimização, tais como [Bulusu et al., 2001], [Wang et al., 2006], [Krishnan, 2004], [Zhang e Arora, 2002], [Cerpa e Estrin, 2004] e [Sohrabi et al., 2000].

3.4 Arquitetura Manna

A arquitetura Manna [Ruiz, 2003] foi proposta para prover uma solução integrada de gerenciamento para diferentes aplicações de RSSFs. A abordagem utilizada no desenvolvimento da arquitetura trata de situações complexas do gerenciamento de RSSF pela decomposição do problema em sub-problemas menores, seguindo refinamentos sucessivos, isto é, trabalhando com cada área funcional e nível de gerenciamento. Além disso, a arquitetura Manna propõe um novo nível de abstração chamado Funcionalidades das RSSF (configuração, sensoramento, manutenção, processamento, comunicação), conforme ilustra a Figura 3.1.

A definição dos serviços de gerenciamento é uma tarefa que consiste em encontrar quais são as funções que devem ser executadas, quando e com quais dados. Os serviços de gerenciamento são executados através de um conjunto de funções as quais precisam ser bem sucedidas para que um dado serviço seja concluído. Funções de gerenciamento representam a menor parte funcional de um serviço de gerenciamento. Os modelos ou mapas de RSSF, definidos pela

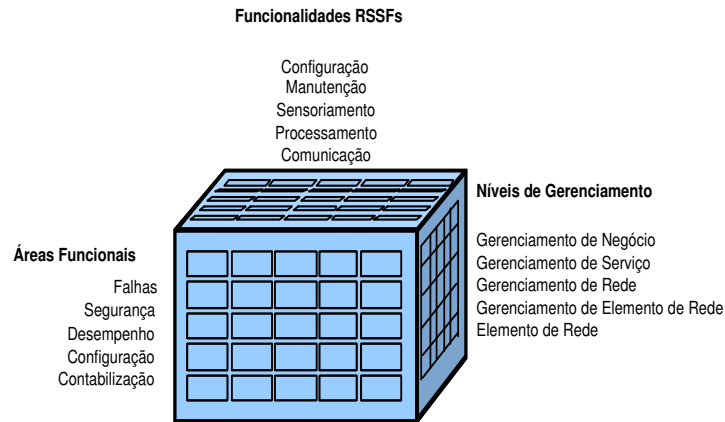


Figura 3.1: Gerenciamento tridimensional proposto pela arquitetura Manna.

arquitetura Manna, representam o estado da rede e servem como uma referência para o gerenciamento. Como exemplos de modelos frequentemente utilizados no gerenciamento de RSSFs pode-se citar mapas de topologia, energia e área de cobertura.

A Figura 3.2 representa um esquema para a construção do gerenciamento, começando pela definição dos serviços e funções de gerenciamento os quais utilizam modelos para alcançar seus objetivos. Um serviço de gerenciamento pode utilizar uma ou mais funções de gerenciamento. Diferentes serviços podem utilizar funções em comum, as quais utilizam modelos para recuperar um estado da rede, considerando um dado aspecto. Os serviços e as funções de gerenciamento utilizam e produzem informações de gerenciamento.

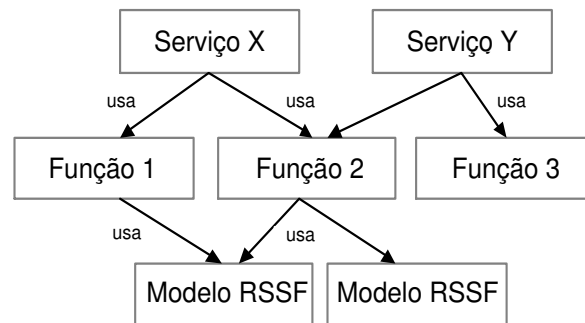


Figura 3.2: Relacionamento entre serviços, funções e modelos de gerenciamento.

A arquitetura Manna foi construída com base no paradigma do auto-gerenciamento, o que permite a definição de RSSFs autônomicas. Essa arquitetura também estabelece uma separação entre as aplicações da rede e seu gerenciamento, através da proposição de três arquiteturas:

Arquitetura de Informação: proposta para garantir soluções comuns para gerenciamento através da definição de um modelo de informação e uma estratégia para se obter informações de gerenciamento. O modelo de informação provê uma base para o entendimento das carac-

terísticas da rede e promove tratamento uniforme para os vários aspectos do gerenciamento de recursos. As três dimensões de gerenciamento devem ser consideradas pela arquitetura na definição do gerenciamento de informações.

Arquitetura Funcional: proposta para o planejamento de possíveis localizações para as entidades de gerenciamento, quais serviços e funções de gerenciamento serão implementados e como eles poderão ser executados. As informações e condições para se executar serviços e funções são obtidas dos modelos de RSSFs definidos pela arquitetura de informação. A escolha do gerenciamento depende das três dimensões utilizadas.

Arquitetura Física: proposta para prover interfaces entre entidades de gerenciamento. Não define uma pilha de protocolos para essas interfaces, mas provê perfis de protocolos que possam ser adequados para cada tipo de aplicação.

O primeiro passo para a construção de uma solução autonômica para redes é a definição da arquitetura de gerenciamento utilizada. Para a solução de RSSFs autonômicas apresentada neste trabalho, a arquitetura Manna foi escolhida. Além de ser pioneira na determinação de serviços e funções para este tipo rede, esta arquitetura já prevê o uso de aspectos autonômicos no gerenciamento dos nós sensores, sendo portanto bastante adequada para utilização em uma solução geral de controle autonômico. Assim, todos os aspectos dessa solução implementados em última instância pelo elemento autonômico são referentes as diretivas de gerenciamento propostas pela arquitetura Manna. A proposta de um ESA para RSSFs pode ser considerada uma extensão ao objetivo da arquitetura Manna, que é o de permitir que os componentes dessas redes gerenciem a si próprios, com o mínimo de intervenção humana.

3.5 Gerenciamento Tradicional

O crescimento e diversificação do número de componentes, aplicações e perfis de usuários das redes de computadores, tornou importante o uso de soluções de gerenciamento que permitem um controle integrado e organizado de diversos parâmetros das mesmas. Em geral o gerenciamento de redes é uma atividade que emprega uma variedade de ferramentas, aplicações e dispositivos para auxiliar gerentes humanos no controle e supervisão de redes. O objetivo principal do gerenciamento é portanto promover a produtividade de recursos e garantir a qualidade dos serviços providos. Basicamente uma solução de gerenciamento é composta por duas entidades: agente e gerente. O agente é um módulo de software que extrai, modela e armazena em uma base de informações de gerência, dados sobre o elemento gerenciado no qual reside. É a parte da aplicação de gerenciamento que atua sobre o elemento gerenciado respondendo às operações do gerente e enviando notificações na ocorrência de algum evento. Ele pode prover essas informações de forma pró-ativa ou reativa para um ou mais gerentes por meio de um protocolo de gerenciamento de redes. O gerente é a entidade responsável por receber informações enviadas por agentes e reagir a elas executando uma ou mais ações, tais

como envio de notificações ao operador da rede, realização de armazenamento (*log*) de eventos e execução de serviços de diagnóstico, recuperação de falhas, dentre outros [Sloman, 1994]. É a parte da aplicação responsável por emitir operações de gerenciamento aos agentes, receber e tratar respostas e processar as informações. Cabe ao gerente toda a inteligência do gerenciamento.

A *International Organization for Standardization* (ISO) [ISO, 2006], órgão responsável pela padronização de vários conceitos ligados ao gerenciamento de redes, definiu um modelo de gerenciamento que contém cinco áreas funcionais de gerência e cinco níveis de gerenciamento, propostos com o objetivo de simplificar o entendimento das principais funções para sistemas de gerenciamento de redes.

As áreas funcionais definidas para modularização e simplificação do entendimento e desenvolvimento de soluções de gerenciamento são [Stallings, 1998]:

- Gerenciamento de Desempenho: quantifica e disponibiliza vários aspectos do desempenho da rede. As três etapas envolvidas com essa área funcional são: aquisição dos dados relevantes, análise dos mesmos e comparação com limites inferiores e superiores para classificação.
- Gerenciamento de Configuração: monitora informações de configuração da rede para que os efeitos de operação com várias versões de elementos de hardware e software possam ser observados e gerenciados.
- Gerenciamento de Contabilização: responsável por quantificar parâmetros de utilização da rede para que o uso individual ou por grupos dos recursos das mesmas possa ser regulado de forma apropriada.
- Gerenciamento de Falhas: detecta, registra e notifica usuários da ocorrência de falhas, podendo corrigi-las sempre que possível. Inicialmente determina sintomas e isola problemas. Em seguida o problema é corrigido e a solução testada. Finalmente, a detecção e resolução do problema são registrados.
- Gerenciamento de Segurança: o objetivo é controlar o acesso aos recursos da rede para que a mesma não sofra ataques de segurança intencionais ou não intencionais e informações sensíveis não sejam acessadas por usuários sem autorização adequada.

Cada uma das áreas funcionais descritas acima pode ser analisada de diferentes formas, de acordo com os níveis de gerenciamento escolhidos. Existem cinco níveis, sendo eles: (1) elemento de rede, (2) gerenciamento de elemento de rede, (3) gerenciamento de rede, (4) gerenciamento de serviço, (5) gerenciamento de negócio. Cada um desses níveis provê uma abstração da rede, direcionando as ações das funções e serviços de gerenciamento das áreas funcionais [Sloman, 1994].

3.5.1 Arquiteturas de Gerenciamento Tradicional

Existem disponíveis na literatura algumas arquiteturas de gerenciamento, que definem o modelo de relacionamento entre agentes e gerentes, isto é, como essas duas entidades devem se comportar para que os serviços e funções de gerenciamento possam ser executados. A seguir estão brevemente descritas algumas dessas arquiteturas:

CMIS (*Common Management Information Services*) [Stallings, 1998]: a função fundamental dentro de um sistema de gerenciamento *Open Systems Interconnection* (OSI) é a troca de informações de gerenciamento entre duas entidades (gerente e agente) por meio de um protocolo. Essa funcionalidade nesses sistemas é conhecida como elemento para serviço de informação comum de gerenciamento (*Common Management Information Service Element* (CMISE)), e é especificada em duas partes: CMIS e *Common Management Information Protocol* (CMIP). O CMIS define os serviços providos para sistemas de gerenciamento OSI. Esses serviços podem ser invocados por processos de gerenciamento para comunicação remota. Os serviços CMIS são especificados em termos de primitivas que podem ser vistas como comandos ou chamadas de procedimentos com parâmetros. Existem dois tipos de serviços: serviços confirmados que requerem que um processo de gerenciamento remoto envie uma resposta para indicar recebimento e sucesso ou falha da operação requisitada; Serviços não confirmados que não provêem respostas. As entidades de gerenciamento da arquitetura CMIS utilizam protocolo CMIP para especificação do formato da unidade de dados do protocolo (*Protocol Data Unit* (PDU)) e procedimentos associados.

SNMP (*Simple Network Management Protocol*) [Stallings, 1998]: arquitetura projetada para ser facilmente implementada como uma ferramenta básica para o gerenciamento de redes, utilizada para se alcançar necessidades de gerenciamento de curto prazo. Dado o lento progresso dos sistemas de gerenciamento OSI, SNMP preencheu as necessidades de gerenciamento dos sistemas atuais e se tornou o esquema de gerenciamento de redes padrão dominante. O SNMP provê um arcabouço para definição de informações de gerenciamento e um protocolo para troca dessa informação. O modelo assume a existência de gerentes e agentes, sendo o primeiro um módulo de software em um sistema de gerenciamento, responsável por gerenciar parte ou todas as configurações para aplicações e usuários. O segundo é um módulo de software em um dispositivo gerenciável responsável por manter informações locais de gerenciamento, entregando-as para um gerente. Uma troca de informações de gerência pode ser iniciada pelo gerente (varredura) ou agente (notificação). O SNMP foi projetado para ser um protocolo da camada de aplicação que seja parte da arquitetura de protocolos TCP/IP, com uso do protocolo UDP na camada de transporte.

TMN (*Telecommunication Management Network*) [TMN, 1992]: é uma arquitetura que serve como modelo genérico de rede de gerenciamento sobre uma rede de telecomunicações,

e que permite a gerência dos diversos equipamentos que compõem essa última. Fornece uma estrutura organizada para interconectar diversos tipos de sistemas de suporte a operações, com troca de informações de gerenciamento através de interfaces padronizadas, incluindo a definição de protocolos e mensagens. Essa arquitetura portanto possui suporte para o gerenciamento de sistemas heterogêneos. Três arquiteturas básicas compõem a TMN: arquitetura funcional (baseada em blocos funcionais, fornece as funções gerais que possibilitam a execução dos procedimentos de gerenciamento), arquitetura de informação (modelo de informação usado no gerenciamento OSI) e arquitetura física (provê meios para implementação dos blocos funcionais definidos pela arquitetura funcional).

Gerenciamento baseado em políticas [Chadha et al., 2002]: arquitetura que utiliza políticas para tratar de situações que possam ocorrer em um sistema, simplificando e automatizando em grande parte a tarefa de gerenciamento. Políticas são regras operacionais que podem ser utilizadas para manutenção de ordem, segurança, consistência, dentre outros aspectos da rede. A aplicação de políticas no contexto de redes com o objetivo de se obter gerenciamento, operação e controle do sistema automatizados caracteriza esse tipo de arquitetura. Uma das principais arquiteturas de gerenciamento baseado em políticas é aquela definida pela *Engineering Task Force (IETF)/Distributed Management Task Force (DMTF)* [Damianou et al., 2002]. O arcabouço de políticas proposto por este grupo é composto por quatro elementos: ferramenta de gerenciamento (MT), permite que administradores definam as políticas a serem utilizadas dentro da rede; Repositório de políticas (PR), utilizado para armazenar políticas geradas pelas ferramentas de gerenciamento; *Policy decision point* (PDP), ponto intermediário que se comunica com o repositório, recupera e interpreta as políticas armazenadas e toma decisões baseadas nestas; *Policy enforcement point* (PEP), dispositivo capaz de aplicar e executar diferentes políticas, recebidas através de comunicação com PDP (ver Seção 4.4).

Gerenciamento baseado em Web: é uma arquitetura em muitos casos pode ser utilizada em conjunto com outras. Os elementos da rede nesse caso possuem entidades de gerenciamento (agentes e gerentes) definidas por outras arquiteturas, tais como SNMP ou CMIS, implementando serviços e protocolos de comunicação de acordo com as especificações dessas arquiteturas. No entanto, a comunicação das entidades com o gerentes, administradores e operadores externos é realizada através da pilha de protocolos Web: protocolos HTML e HTTP na camada de aplicação, utilizando arquitetura TCP/IP. Um agente *proxy* que possua um servidor Web e um gerente capaz de se comunicar utilizando protocolo implementado pelas entidades de gerenciamento da rede, são responsáveis por fazer a ligação dos elementos da rede e o gerente humano. Existe também a possibilidade de que os próprios elementos da rede possuam um servidor web embutido, sendo portanto capazes de se comunicar diretamente com o gerente humano externo [Ju et al., 2000].

Web Services: define uma arquitetura genérica baseada em tecnologia *eXtensible Markup Language* (XML) que possui suporte para vários fabricantes e disponível em diversas plataformas. Consiste basicamente de componentes de aplicações independentes, disponibilizados na Web de forma que outras aplicações possam encontrá-los e utilizá-los. Essa arquitetura possui três elementos principais responsáveis por hospedar e publicar *Web Services*, manter registros sobre os *Web Services* publicados e pesquisar *Web Services*, requisitando-os uma vez encontrados. Características tais como flexibilidade, computação distribuída, e facilidade de utilização tornam *Web Services* uma arquitetura de gerenciamento atrativa. Uma outra característica importante dessa arquitetura é a possibilidade de se construir *Web Services* maiores e mais complexos a partir da combinação e reuso de outros mais simples, permitindo a integração de serviços de gerenciamento de redes ou mesmo a criação de novos serviços [A. Pras and T. Drevers and R. Meent and D. Quartel, 2004, Fioreze et al., 2005].

As arquiteturas mencionadas acima diferem da proposta do paradigma da computação autonômica por estarem baseadas no uso das duas entidades de gerenciamento (agentes e gerentes) e por manterem o foco da execução do controle e supervisão em um gerente humano externo. Dessa forma, os gerentes da rede devem sempre requisitar informações e enviar ordens de ações para que os agentes realizem suas tarefas, e também solicitar ordens e enviar dados, informações e relatórios para o gerente humano externo, agindo sempre de acordo com os comandos dados por esse último. Em uma rede autonômica, cada elemento possui um gerente embutido (entidade de controle de um elemento autonômico), capaz de executar ações de forma independente e pró-ativa. O auto-gerenciamento se mostra mais adequado para as RSSFs devido às características específicas desse tipo de rede e seus componentes, conforme descritas pelas Seções 1.1 e 1.2.

3.6 Considerações Finais

Esse capítulo apresentou uma série de trabalhos considerados de alguma forma relacionados àquele apresentado nessa dissertação. Os trabalhos foram divididos em categorias, de acordo com tipo de sistema ou rede a que se aplicam. Conforme foi dito, existem vários softwares comerciais para computação autonômica desenvolvidos por grandes empresas da área de TI, tais como IBM, SUN e Microsoft, aplicáveis em sistemas de TI tradicionais. Além disso, também podem ser encontrados diversos trabalhos de pesquisa publicados na literatura que descrevem soluções autonômicas para redes tradicionais com fios. As soluções propostas nos dois casos acima não se aplicam para o tipo de rede considerado nesse trabalho, visto que não consideram as características e demandas particulares das mesmas e dos elementos que as compõem. Em seguida foram apresentadas soluções encontradas na literatura que propõem a implementação de serviços autonômicos para redes sem fio e RSSFs em particular. Nenhum desses trabalhos apresenta a proposta de um elemento autonômico específico para nós senso-

res, e nem mesmo um esquema para implementação integrada e dinâmica de vários serviços autonômicos, possuindo assim objetivos diferentes daqueles apresentados para essa dissertação. Em todos os casos, o foco dos trabalhos está na implementação de uma funcionalidade, serviço ou função em particular, e não na definição integrada dos mesmo em um elemento autonômico aderente ao elemento de rede considerado.

A arquitetura Manna para o gerenciamento específico e integrado de RSSFs, a qual é utilizada como uma das bases para elaboração do elemento autonômico para RSSF, também foi apresentada. Finalmente foram discutidos alguns aspectos importantes sobre o gerenciamento tradicional de redes e as principais soluções relacionadas, tais como SNMP, PBNM, TMN, dentre outros. O capítulo a seguir apresenta o elemento sensor autonômico definido para RSSFs e discute em detalhes a implementação de cada um de seus serviços e da base de conhecimento.

Capítulo 4

Elemento Sensor Autônômico

Este capítulo² apresenta o modelo de um elemento autônômico para RSSFs, chamado Elemento Sensor Autônômico (ESA), capaz de ser executado em plataformas com restrições severas de hardware, software, comunicação de dados e energia, e ainda assim promover a produtividade destes recursos e a qualidade dos serviços provisionados pelos elementos da rede. Conforme descrito na Seção 2.5, os EAs podem ser vistos como a menor parte de um sistema autônômico e, apesar de terem seus papéis de gerentes bastante definidos, as tarefas a serem desempenhadas e a forma como as mesmas são implementadas diferem de um tipo de rede ou sistema para outro. Em redes de computadores tradicionais por exemplo, um EA pode ser responsável por diversas funções de gerenciamento, implementando-as de forma bastante sofisticada e complexa. Por outro lado, uma vez que EAs em RSSFs estão embutidos em dispositivos com recursos limitados, eles precisam ser robustos, flexíveis, dinâmicos e devem observar rigorosamente o compromisso entre complexidade, precisão e disponibilidade de recursos. A Seção 4.1 introduz o conceito do ciclo autônômico de controle infinito para o contexto das RSSFs. Os serviços de monitoração, análise, planejamento e execução propostos para o ESA estão descritos respectivamente nas Seções 4.2, 4.3, 4.4 e 4.5. A Seção 4.6 apresenta os componentes de uma base de conhecimento proposta nesse trabalho para ESAs e a Seção 4.8 apresenta as considerações finais para o capítulo.

4.1 Visão Geral do Gerente Autônômico de um ESA

De acordo com o formato de um EA genérico proposto em [Kephart e Chess, 2003], um elemento autônômico é constituído de cinco partes: monitoração, análise, planejamento, execução e uma Base de Conhecimento (BC). No modelo do ESA proposto neste trabalho e ilustrado na Figura 4.1, cada uma das partes do ciclo são serviços de gerenciamento do ESA, executados de forma autônômica através de um conjunto de funções. A BC pode apresentar diferentes componentes, dentre eles uma base de informações de gerenciamento (*Management*

²O conteúdo desse capítulo foi parcialmente publicado como artigo completo no Primeiro Simpósio Latino Americano de Computação Autônômica, 2006 [Braga et al., 2006b] (em inglês) e no XXIV Simpósio Brasileiro de Redes de Computadores, 2006 [Braga et al., 2006c] (em português).

Information Base (MIB)) local, uma base de informações da aplicação para a qual a RSSF foi projetada (*Application Information Base* (AIB)), um módulo que corresponde ao protocolo de gerenciamento utilizado pelos nós sensores e uma máquina de políticas (ver Seção 4.6).

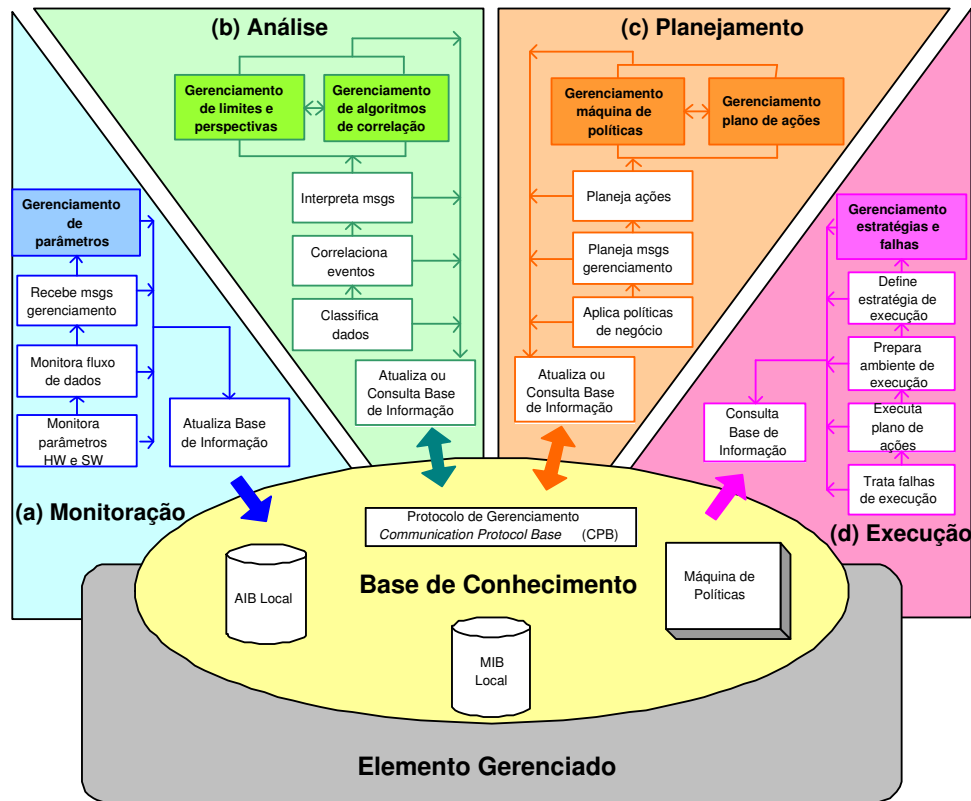


Figura 4.1: Visão geral dos serviços, funções e componentes de conhecimento que compõem o Elemento Sensor Autônomo.

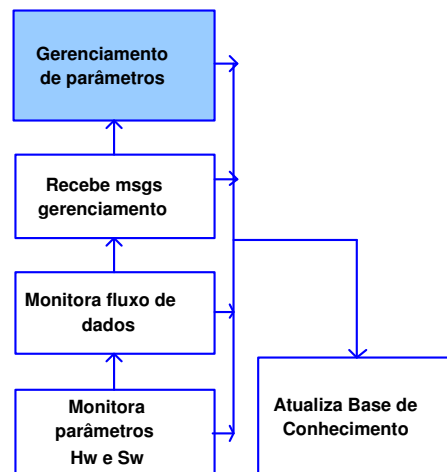
Os serviços são conectados através de um laço autônomo de execução infinito. Todos os serviços interagem de alguma forma com a BC. A seguir estão descritos individualmente cada um dos serviços providos pelo ESA e suas respectivas funções, além da BC e seus componentes. As funções da Figura 4.1 que prevêem algum tipo de gerenciamento (gerenciamento de parâmetros, limites, perspectivas, máquina de políticas, plano de ações, estratégias e falhas) indicam pontos em que algoritmos de aprendizagem podem ser incluídos. Embora existam limitações físicas, algoritmos simples e eficientes podem ser utilizados e o ESA pode decidir executá-los ou não de acordo com a disponibilidade de recursos.

4.2 Serviço de Monitoração

O serviço de monitoração (Figura 4.2) de um elemento autônomo é responsável pela implementação do auto-conhecimento e auto-consciência (Seção 2.5.3.1). Tal serviço pode ser projetado de diversas maneiras, dependendo do sistema autônomo ao qual pertence. Em redes tradicionais por exemplo, vários parâmetros de hardware, software e do ambiente podem ser monitorados e em seguida combinados e armazenados na base de conhecimento. Além

disso a frequência de monitoração pode não ser um problema, uma vez que computadores tradicionais podem processar de forma eficiente tarefas concorrentes. Por outro lado, o contexto das RSSFs é bastante diferente do das redes tradicionais. Uma vez que os componentes das RSSFs possuem grande limitação de recursos, dentre eles energia, memória e processamento, cada serviço que o ESA embutido possa prover deve ser cuidadosamente planejado e gerenciado. Em particular, o serviço de monitoração deve se ater a um conjunto mínimo de parâmetros que represente de forma satisfatória e momentânea o estado do elemento gerenciado, seus componentes e seu ambiente. Cada parâmetro pouco relevante representa um consumo de memória desnecessário.

É importante ressaltar que o conjunto de parâmetros monitorados pelo ESA é extremamente dependente da aplicação para a qual a RSSF foi projetada, assim como do contexto ou momento no qual a rede se encontra. Por exemplo, um conjunto de parâmetros importantes para uma RSSF que monitora um corpo humano pode não ser adequado para outra que realize detecção de focos de incêndio em uma floresta. Além disso, os parâmetros relevantes para uma RSSF que passa por um momento crítico, como falta de recursos ou grande detecção de eventos relevantes, podem não ser os mesmos para a mesma rede no início de suas operações. Assim, este conjunto deve ser definido de maneira independente para cada ocasião.



(a) Monitoração

Figura 4.2: Funções que compõem o serviço de monitoração de um ESA.

Os parâmetros monitorados pelo ESA podem ser divididos em duas categorias: internos e externos. Parâmetros internos estão ligados ao conceito do auto-conhecimento e refletem o estado atual do hardware e software do elemento gerenciado. Parâmetros externos são aqueles ligados à auto-consciência e constroem a visão do elemento gerenciado em relação às condições da rede, suas conexões com vizinhos e o estado destes.

- **Função Monitora Parâmetros de Hardware e Software:** a coleta dos parâmetros internos é realizada por funções de acesso direto que o ESA possui aos componentes de hardware e software do elemento gerenciado. O hardware da plataforma de nó sensor

utilizada (ver Seção 1.2) deve permitir acesso aos seus componentes, para que os mesmos possam ser monitorados. Alguns componentes das plataformas existentes atualmente, como Mica Motes 2 ou MicaZ [Hill e Culler, 2002], atendem a estas condições.

- **Função Monitora Fluxo de Dados e Função Recebe Mensagens de Gerenciamento:** em relação aos parâmetros externos, duas funções principais devem ser implementadas: monitoração do fluxo de dados e recepção de mensagens de gerenciamento. As RSSFs, quaisquer que sejam suas aplicações e formas de sensoriamento e disseminação de dados, produzem fluxos de dados coletados. Esses fluxos são disseminados em um esquema de comunicação multi-saltos até o PA da rede. Assim, cada nó sensor da rede não só produz um fluxo de dados, como é responsável por disseminar outros fluxos, criados por outros nós da rede, e que passam por ele para alcançar o PA. A análise dos múltiplos fluxos de dados que são criados ou passam por um nó sensor, pode fornecer parâmetros externos importantes para a auto-consciência do ESA. Além disso, em algumas ocasiões, os diversos ESAs presentes na rede podem desejar trocar mensagens de gerenciamento entre si ou receber mensagens vindas do PA. Dentre alguns motivos para envio de mensagens estão negociação de serviços e troca de informações. Neste caso, o ESA deve monitorar o recebimento destas mensagens.

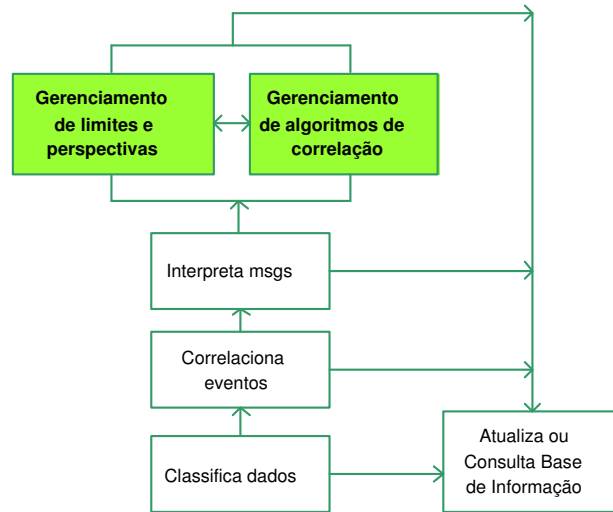
Em [Elnahrawy, 2003] são discutidos a importância e os desafios da análise de fluxo de dados em RSSFs. Segundo o autor, existem importantes diferenças entre fluxo de dados sensoriados e fluxos de dados tradicionais: o fluxo de dados sensoriados é apenas uma amostra do conjunto de dados como um todo. A taxa de amostragem varia de aplicação para aplicação. Os dados coletados nestes fluxos são em geral imprecisos e contém ruídos, e a aquisição de dados em RSSF custa caro em termos de consumo de energia. Os dados do fluxo de uma RSSF se tornam sem utilidade se seus prazos são perdidos. Assim, sistemas de análise de fluxos tradicionais não são aplicáveis para fluxos de dados sensoriados, o que leva à conclusão de que o tratamento destes requer sistemas especiais adaptados às limitações das RSSFs.

- **Função Atualiza Base de Conhecimento e Função Gerenciamento de Parâmetros:** ao final da execução deste serviço, o ESA deve atualizar a BC e definir quais parâmetros internos e externos serão mantidos no conjunto de parâmetros de monitoração. Os dados inseridos na BC serão utilizados pelo próximo serviço do laço autônomo.

4.3 Serviço de Análise

A verificação dos contextos interno e externo de um elemento autônomo será realizada pelo serviço de análise (Seção 2.5.3.2). É responsabilidade deste serviço transformar os dados de entrada obtidos pelo serviço de monitoração, em informações úteis que serão analisadas, e que levem a conclusões sobre aspectos de desempenho, produtividade e cumprimento dos níveis de qualidade de serviço desejados e de acordos de negociação de serviço estabelecidos por exemplo, para diferentes áreas funcionais, níveis de gerenciamento e considerando-se uma ou

mais funcionalidades das RSSFs. As funções definidas para o serviço de análise do ESA podem ser vistas na Figura 4.3. O conceito autônomo ligado a este serviço é o auto-diagnóstico.



(b) Análise

Figura 4.3: Funções que compõem o serviço de análise de um ESA.

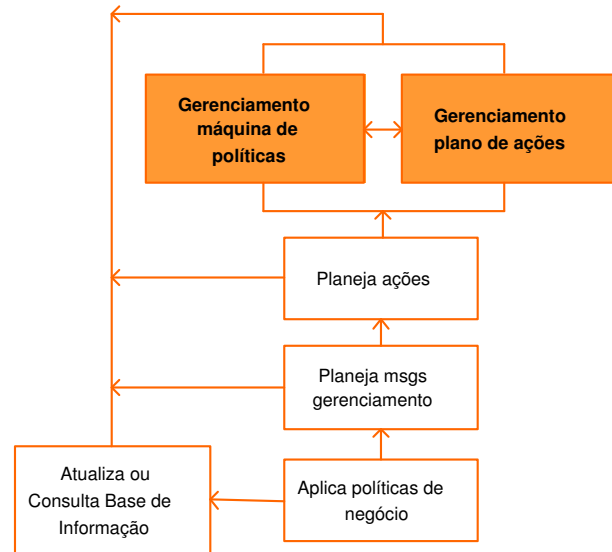
- **Função Classifica Dados:** até o momento, o ESA possui novos dados, mas não novas informações, com as quais poderia analisar a situação do elemento gerenciado. A primeira função a ser executada é a classificação dos dados, transformando-os em eventos. É necessário que o ESA possua valores limites (*thresholds*) armazenados em sua BC, com os quais classificará os dados em eventos utilizando alguma escala. O serviço pode por exemplo, utilizar alguns limites aplicáveis a vários tipos de dados.
- **Função Gerenciamento de Limites:** os limites devem ser ajustados sempre que necessário, e ainda aqueles que não são mais utilizados podem ser descartados. Cabe ao próprio serviço de análise realizar o gerenciamento destes limites. O ESA pode adotar várias estratégias para verificação da validade dos limites utilizados, tais como percepção da eficiência das ações executadas, isto é, verificar se as ações executadas estão melhorando os aspectos necessários da forma desejada, observação de uma classificação de dados desproporcional, dificuldade em classificar os dados nos limites disponíveis, dentre outros.
- **Função Correlaciona Eventos:** em seguida, a função de correlação dos eventos identificados será executada. Existem vários algoritmos de correlação disponíveis na literatura. Alguns deles são considerados simples, tais como a correlação baseada em regras, enquanto outros podem ser bastante complexos (redes bayesianas ou redes neurais). O uso de determinado algoritmo será definido pelo ESA com base na adequação do mesmo à situação atual do elemento gerenciado. O compromisso entre complexidade e precisão deve ser considerado e analisado. Algoritmos mais sofisticados poderiam levar a

conclusões precisas, completas e seguras. No entanto, os custos computacionais e de consumo de energia associados podem ser altos demais para as condições atuais do nó sensor. Além disso, pode não haver mesmo a necessidade de obter uma conclusão tão precisa, ou seja, aproximações podem levar a resultados muito próximos, a um custo mais acessível.

- **Função Gerenciamento de Algoritmos de Correlação:** caso julgue necessário e viável, o ESA pode manter mais de uma opção de algoritmo armazenados, utilizando-os nos momentos oportunos. Ele pode também descartar opções que já não são mais utilizadas e identificar a necessidade de requisitar o recebimento de um novo código, contendo um algoritmo mais adequado. O gerenciamento dos algoritmos de correlação é mais uma função de responsabilidade do serviço de análise.
- **Função Interpreta Mensagens:** uma vez que o ESA pode ter recebido mensagens de gerenciamento de outros ESAs da rede ou mesmo do PA, as quais foram recebidas pelo serviço de monitoração, ele deve conter uma função de interpretação dessas mensagens. O serviço de análise contém esta função, e utiliza conhecimentos sobre o protocolo de gerenciamento utilizado pela rede armazenados na BC, para executá-la. De acordo com o tipo e conteúdo das mensagens, novas classificações e correlações podem ser necessárias.
- **Função Gerenciamento de Perspectivas:** uma importante discussão que surge neste ponto diz respeito a quais aspectos do elemento gerenciado serão observados e em quais níveis, ou seja, quais áreas funcionais e níveis de gerenciamento serão considerados. Além disso, como o modelo do ESA está baseado no gerenciamento tridimensional proposto pela arquitetura Manna, é ainda necessário definir quais funcionalidades das RSSFs devem ser analisadas. A Figura 3.1 da Seção 3.4 mostra o cubo proposto pela arquitetura Manna que sugere as interseções entre as três dimensões de gerenciamento utilizadas por essa arquitetura. Cada interseção define uma célula, que é na verdade uma perspectiva sobre o gerenciamento de uma RSSF que indica área(s) funcional(ais), nível(is) de gerenciamento e funcionalidade(s) consideradas. Assim sendo, o ESA deve definir quais perspectivas serão consideradas. Podem ser escolhidas uma ou um conjunto de perspectivas, de acordo com cada situação em particular. Além disso, este conjunto pode ser modificado sempre que necessário.
- **Função Atualiza ou Consulta Base de Conhecimento:** finalizando a etapa de análise, uma função de atualização da BC deve ser executada. Esta função atualizará ou mesmo incluirá valores secundários na MIB local e, além disso (re)construirá mapas relativos ao seu ambiente. Neste momento o ESA já possui tudo o que precisa para planejar quais ações devem ser tomadas. É exatamente isso que faz o serviço de planejamento apresentado a seguir pela Seção 4.4.

4.4 Serviço de Planejamento

O serviço de planejamento (Seção 2.5.3.3) de um EA possui como principal objetivo escolher ou elaborar um plano de ações a serem seguidas, de acordo com o resultado da análise realizada pelo serviço anterior. Mais uma vez, um conjunto de funções associadas a este serviço pode ser definido de acordo com o grau de precisão e autenticidade desejado e o nível de complexidade permitido. A Figura 4.4 ilustra o serviço de planejamento do ESA.



(c) Planejamento

Figura 4.4: Funções que compõem o serviço de planejamento de um ESA.

- **Função Aplica Políticas de Negócio:** inicialmente, uma função busca na BC as informações ou conclusões obtidas sobre o conjunto de perspectivas consideradas pelo serviço de análise. Essas conclusões serão entregues para a função que vai aplicá-las às políticas correspondentes. As políticas vão mapear as conclusões em uma ou mais ações, as quais serão executadas posteriormente com o intuito de ajustar, melhorar ou consertar algo, em uma ou mais perspectivas do ESA.
- **Função Planeja Ações:** identificadas as ações, uma função será responsável por organizá-las em um conjunto, montando assim o plano de ações do ESA. As ações utilizadas pelo ESA podem estar relacionadas a vários conceitos autônômicos, dentre eles auto-recuperação, auto-otimização, auto-proteção, auto-organização, auto-manutenção, auto-sustento e auto-serviço [Ruiz, 2003] (ver Seção 2.5).
- **Função Planeja Mensagens de Gerenciamento:** caso seja identificada pelas funções acima a necessidade de se enviar mensagens de gerenciamento para o gerente externo, ou para outros ESAs da rede, essa função planejará tais mensagens. Tais mensagens podem ser planejadas com intuito de alertar sobre ocorrência de eventos, negociar

ou solicitar recursos externos ou de outros ESAs, divulgar informações e dados de configuração e disponibilidade de recursos, dentre outras possibilidades.

- **Função Gerenciamento Máquina de Políticas e Gerenciamento de Ações:** o gerenciamento da máquina de políticas e das ações que o ESA pode executar são responsabilidades também do serviço de planejamento. Neste caso, ele deverá sempre manter um conjunto adequado de políticas e ações, descartar os itens desnecessários e identificar o momento em que atualizações são necessárias. Especificamente em relação às políticas, a função de gerenciamento da máquina deve ser capaz de receber políticas de negócio, que são políticas de alto nível especificadas pelo observador da rede, e transformá-las em políticas mais simples ou mesmo SLAs (*Service Level Agreements*) diretamente aplicáveis.
- **Função Atualiza ou Consulta Base de Conhecimento:** todas as funções acima podem recorrer a BC para consultar algum dado ou informação. Em particular, a função de planejamento de mensagens de gerenciamento recorre às especificações sobre o protocolo de gerenciamento utilizado para organizar a montagem das mesmas. Além disso, todos os novos conhecimentos, novas ações ou políticas são inseridas por este serviço na BC do ESA e aqueles que não são mais válidos ou necessários são descartados.

Neste ponto, as principais funções autonômicas do ESA já foram realizadas. O plano de ações será utilizado pelo próximo e último serviço do laço autonômico, chamado serviço de execução.

4.5 Serviço de Execução

O serviço de execução (Seção 2.5.3.4) está relacionado com um dos principais aspectos do paradigma da computação autonômica: a auto-configuração. É através deste aspecto que o EA consegue atuar sobre o hardware ou software do elemento gerenciado, buscando configurá-los da melhor maneira possível para as necessidades deste elemento e às condições do ambiente que o cercam. Não é função deste serviço analisar se o plano de ações está ou não correto ou se as ações são aplicáveis.

- **Função Consulta Base de Conhecimento:** no modelo de ESA proposto neste trabalho, conforme mostra a Figura 4.5, o primeiro passo do serviço de execução é consultar a BC para recuperar o plano de ações definido pelo serviço de planejamento. De acordo com as perspectivas analisadas e o plano de ações escolhido, a auto-configuração pode ser realizada em diferentes níveis de gerenciamento, uma ou mais áreas funcionais e uma ou mais funcionalidades de RSSFs.
- **Função Prepara Ambiente de Execução:** organiza e estrutura o ambiente de acordo com as ações. Pode ser necessário recuperar dados ou informações, preparar determinado hardware, executar algum componente de software, dentre outras atividades.

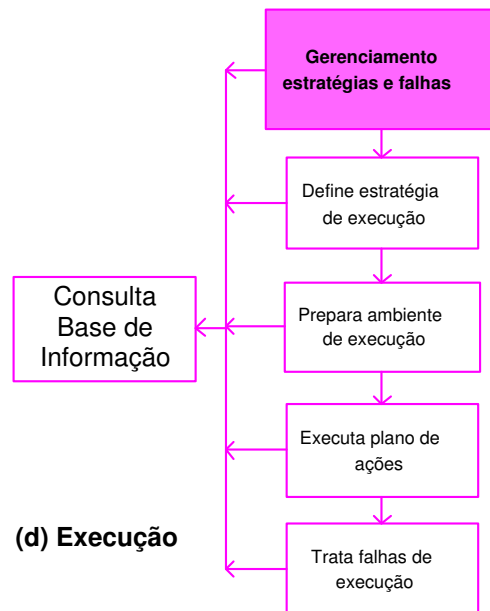


Figura 4.5: Funções que compõem o serviço de execução de um ESA.

- **Função Define Estratégia de Execução:** antes de chamar a função que efetivamente executa o plano de ações, uma outra função responsável por definir a melhor estratégia de execução é invocada. A estratégia é o mapeamento entre as ações e os correspondentes passos a serem executados. Principalmente no caso em que uma ação possa interferir no resultado de outra, é importante que o ESA identifique qual a melhor estratégia de execução para as ações de um plano.
- **Função Executa Plano de Ações:** finalmente uma função executa cada ação na ordem em que foram planejadas. Uma das possíveis ações planejadas é o envio de mensagens de gerenciamento. Nesse caso, o ESA deve recorrer as informações sobre o protocolo de gerenciamento utilizado, disponíveis na BC, para elaborar a mensagem e em seguida enviá-la. Se as execuções forem todas bem sucedidas, o ESA completa uma iteração do laço autônomo. As mudanças realizadas devem causar algum impacto sobre o elemento gerenciado, o ambiente no qual ele se encontra e/ou em seus vizinhos. Estas mudanças serão percebidas pelo ESA, através do serviço de monitoração, na próxima iteração do laço.
- **Função Trata Falhas de Execução:** caso uma ou mais execuções apresentem problemas, o ESA deve invocar uma função para identificação e tratamento de falhas. Esta função procurará formas de contornar os problemas ocorridos. Mecanismos sofisticados de identificação e correção de falhas podem levar a bons resultados, mas o consumo de recursos associados seria proibitivo. É interessante que esta função implemente estratégias simples e eficientes.
- **Função Gerenciamento de Estratégias e Falhas:** todos os serviços do laço autônomo possuem funções de gerenciamento, que além de controlar e supervisionar aspectos

importantes permite melhoria do desempenho e aprendizado para o ESA. Assim, também o serviço de execução possui uma função responsável por gerenciar as estratégias de execução do plano de ações e também dos tipos de falhas reconhecidas e seus respectivos tratamentos. Essas funções podem decidir pela permanência ou não de itens na BC e executar algoritmos que permitam aprendizado no contexto específico em que atuam.

Todos os serviços descritos acima possuem funções de gerenciamento que definem pontos onde algoritmos de controle, supervisão e aprendizado de máquina podem ser executados. As funções de aprendizado podem ser executadas sempre que a produção de um conhecimento for viável ou necessária.

4.6 Base de Conhecimento (BC)

A BC (Seção 2.5.3.5) armazena dados, informações, políticas, limites, dentre outros itens. O principal compromisso observado pelo ESA neste caso é entre consumo de espaço de armazenamento e precisão. Os nós sensores possuem uma quantidade limitada de espaço em memória. Além disso, a restrição no consumo de energia requer que mesmo as operações de leitura e escrita em memória sejam bem planejadas. Por outro lado, é interessante que o ESA possa contar com todos os itens necessários em sua BC. O ESA deve portanto manter um conjunto adequado, contendo apenas os itens realmente necessários. Este conjunto pode variar durante o tempo de vida da rede. Assim, os itens desnecessários são descartados e outros novos, se for o caso, inseridos. Conforme ilustrado pela Figura 4.1, a BC possui quatro componentes: uma base de informações de gerenciamento local (MIB - *Management Information Base*), uma base de informações da aplicação (AIB), uma máquina de políticas e um módulo correspondente ao protocolo de gerenciamento utilizado (*Communication Protocol Base* (CPB)). A natureza dinâmica é a principal característica da BC de um ESA.

MIB local: uma base de informação de gerenciamento (MIB) contém um conjunto de informações, organizadas hierarquicamente de acordo com o protocolo de gerenciamento utilizado. Ela define as variáveis simples ou compostas, que podem ser controladas pela aplicação de gerenciamento. A MIB de um EA não é estática, ou seja, seus componentes devem ser constantemente gerenciados e renovados caso haja necessidade [Perkins e McGinnis, 1996]. No caso do ESA, sua MIB deve ser dinâmica e enxuta. Apenas os componentes realmente importantes para o contexto atual da RSSFs devem ser mantidos. Os itens obsoletos, não utilizados durante determinado espaço de tempo são descartados pelo ESA. Uma MIB para RSSFs foi proposta em [Silva et al., 2005]. Tanto dados como informações coletadas ou produzidas e utilizadas pelos serviços de monitoração, análise, planejamento e execução, e que estão relacionadas com características dos elementos de rede, rede, serviços ou negócio, são armazenadas na MIB do ESA. A MIB armazenará limites, mapas, algoritmos, perspectivas, estratégias, falhas, além do conjunto e do plano de ações.

AIB local: é uma base de informações que apresenta estrutura de armazenamento e representação de dados similar a uma MIB, mas que possui apenas conteúdo relacionado com a aplicação para a qual a RSSF foi projetada. Dado que as RSSFs são redes dependentes da aplicação, é importante que a entidade de gerenciamento, conheça e utilize detalhes da mesma, para realizar um controle satisfatório e coerente dos elementos gerenciados. A AIB deve conter apenas limites e representações de dados, tais como mapas, específicos da aplicação.

Máquina de políticas: o gerenciamento baseado em políticas (PBNM) é fundamental para o paradigma da computação autônoma (Seção 4.4). Em qualquer contexto em que este paradigma for aplicado, deverá ser implementado suporte necessário para o uso de PBNM. Assim, uma máquina de políticas é algo que deve fazer parte da BC de um EA. Esta máquina é responsável por armazenar e gerenciar as políticas de negócio especificadas pelo projetista da rede [Chadha et al., 2002]. Além disso, ela transformará tais políticas em outras mais simples ou mesmo em SLAs (*Service Level Agreement*).

A máquina de políticas do ESA, armazenada em sua BC, pode ser utilizada por todos os serviços em suas funções de gerenciamento e em particular pelo serviço de planejamento para definição das ações necessárias, de acordo com resultado do serviço de análise. Ela deve ser uma versão compacta das máquinas de políticas utilizadas em redes tradicionais. Somente os elementos mais necessários, implementados de maneira simples, devem ser mantidos. Assim também devem ser as próprias políticas, algoritmos e SLAs utilizados. Em primeiro lugar, o conjunto de políticas de negócio deve ser pequeno e simples. O mapeamento entre estas políticas e os SLAs ou políticas mais simples deve ser feito através de um algoritmo eficiente, que não demande um grande consumo de recursos. Em última instância, o conjunto de SLAs e políticas diretamente aplicáveis (baixo nível) deve ser pequeno e reaproveitável, ou seja, um ou mais itens deste conjunto pode ser utilizado no mapeamento de uma ou mais políticas de negócio. Somente itens realmente importante e que são freqüentemente utilizados devem permanecer no conjunto. Novos itens podem ser adicionados, caso o ESA perceba que o conjunto atual não seja mais útil para o gerenciamento do nó sensor, dadas as condições atuais da RSSF e o ambiente que a cerca.

Protocolo de comunicação: os ESAs que gerenciam cada um dos nós sensores da rede possivelmente trocarão mensagens de gerenciamento entre si e/ou com o PA em algum(ns) momento(s). Alguns dos motivos que levam à necessidade de comunicação entre ESAs são divulgação de serviços que podem prestar e serviços requisitados, negociação de serviços, troca de dados e informações, reconhecimento da vizinhança, formação de grupos (no caso de rede hierárquicas), dentre outros. Neste caso, estes elementos devem compartilhar uma maneira de trocarem mensagens. A especificação desta maneira é dada pelo protocolo de gerenciamento utilizado. O ESA deve armazenar em sua BC, de uma forma compacta, as especificações do protocolo de comunicação utilizado na rede. Assim, ele pode conhecer a forma de interpretar e criar mensagens. Caso o protocolo utilizado seja atualizado ou mesmo modificado, caberá

ao ESA se adaptar à nova situação.

Os trabalhos descritos em [Silva et al., 2005, Lim, 2006], definem protocolos de gerenciamento para RSSFs. O projeto do ESA prevê que estes elementos deverão utilizar algum protocolo deste tipo, possuindo para tal uma especificação em sua base de conhecimento. Os trabalhos que exploram propostas de protocolos específicos para o gerenciamento de RSSF são complementares a este pois oferecem diferentes opções que podem ser utilizadas de acordo com cada instância do projeto de ESA.

Conforme mencionado anteriormente no Capítulo 2, os gerentes autônomicos de um EA trocam mensagens entre si para a realização de um gerenciamento distribuído sobre a infraestrutura da rede ou sistema, através de interfaces padronizadas chamadas *touchpoints*. O uso de *touchpoints* permite que ESAs instalados em diferentes nós sensores possam se comunicar e estabelecer um ciclo de relacionamento de forma única e padronizada, independente das particularidades de cada plataforma. Nesse caso, as informações sobre os *touchpoints* utilizados para comunicação entre os ESAs da rede também devem ser armazenadas na BC, sendo conhecidas e utilizadas pelo módulo responsável pelo protocolo de gerenciamento.

4.7 Outros Aspectos

Localização de ESAs em RSSFs: de acordo com a definição da arquitetura Manna, uma solução de gerenciamento para uma RSSF deve atender às particularidades desta, apresentando um conjunto de serviços e funções necessários e compatíveis com a disponibilidade de recursos e objetivos de controle e supervisão. Algumas RSSFs possuirão uma solução mínima de gerenciamento, enquanto outras podem apresentar implementações mais elaboradas e completas. Em todos os casos, quando uma solução autônoma qualquer for elaborada para uma RSSF, um ESA associado deve ser instanciado e embutido nos elementos dessas redes. Nesse ponto, deve também ser decidida a localização desses ESAs. Conforme mencionado anteriormente, o modelo de ESA proposto nesse trabalho foi construído com base na arquitetura Manna. Dessa forma, a localização de ESAs na RSSF deve seguir a proposta da arquitetura funcional desenvolvida por essa arquitetura, a qual define diferentes possibilidades de distribuição das entidades de gerenciamento na rede.

A arquitetura Manna trabalha com a visão de dois tipos de entidade de gerenciamento: agentes e gerentes. Os agentes podem ser vistos como um ESA bastante simples, capaz de receber e enviar informações para um gerente, através de um determinado protocolo de gerenciamento e realizar poucas funções de controle e supervisão. Já o gerente é a entidade capaz de executar serviços e funções diversas, implementando, caso seja viável e necessário, uma instância de ESA completa. De acordo com a arquitetura funcional da arquitetura Manna, as entidades de gerenciamento podem ser distribuídas de diferentes formas, inclusive de acordo com a organização da rede (plana ou hierárquica). ESAs completos (gerentes) podem estar distribuídos em determinados pontos de uma rede plana ou em líderes de grupos nas redes hierárquicas. Além disso, eles podem estar implementados nos PAs da rede e também podem aparecer em entidades externas à RSSF. Os ESAs mais enxutos (agentes) podem estar

distribuídos em todos os demais nós da rede ou apenas em pontos específicos, como os próprios líderes de grupo. Em termos de localização portanto, é possível se ter diferentes instâncias de ESAs em diferentes distribuições de uma RSSF, de acordo com a organização, configuração, objetivos de gerenciamento e disponibilidade de recursos da mesma.

Interfaces ESA: três tipos de interfaces (*touchpoints*) devem ser definidos na construção de um modelo de EA específico para determinado tipo de rede ou sistema: interfaces de comunicação entre EAs, interfaces entre os serviços do laço de controle e interfaces entre o ESA e o hardware e software do elemento de rede ou componente de sistema. A definição desse último tipo de interface, conforme mencionado anteriormente, foi considerada fora do escopo desse trabalho. Nesse caso, apenas assumiu-se que era possível acessar hardware e software dos nós sensores que possuíam uma instância de ESA embutida. Para o primeiro e segundo tipos de interface, as implementações definidas pelo modelo apresentado nesse trabalho são respectivamente o protocolo de gerenciamento utilizado e o acesso compartilhado a BC.

4.8 Considerações Finais

Este capítulo apresentou o modelo de um elemento autônomo para RSSFs chamado Elemento Sensor Autônomo (ESA). A elaboração desse modelo e suas particularidades, tais como formato, serviços, funções que compõem cada serviço e conteúdo para a base de conhecimento, é a principal contribuição dessa dissertação. Cada serviço que compõe o formato genérico escolhido como base para o modelo de ESA foi definido em uma seção particular, discutindo-se todas as funções que podem ser implementadas para os mesmos. Cabe ressaltar que nem todas as funções propostas para cada serviço são obrigatórias. Instâncias do modelo proposto podem ser criadas, contendo apenas as funções mais adequadas em cada serviço para o objetivo do gerenciamento. Além disso, os algoritmos utilizados na implementação de cada função podem ser de complexidades computacionais variadas, de acordo com a precisão desejada e quantidade de recursos disponíveis. A base de conhecimento para o ESA também foi definida, considerando-se os elementos que a compõem. A definição dos serviços, funções e elementos da base de conhecimento foi realizada considerando-se as particularidades das RSSFs, seus componentes e aplicações, bem como a definição de gerenciamento para esse tipo específico de redes proposta pela arquitetura Manna.

A utilização de uma instância de ESA embutida em nós sensores pode representar um consumo de memória adicional, sendo que a implementação dos serviços do ciclo autônomo e base de conhecimento deve ser robusta, observando-se sempre o limite de memória imposto pela restrição de recursos computacionais desses elementos de rede. O gerenciamento realizado de forma autônoma e específica, com cada ESA buscando atender às necessidade de cada nó sensor em particular e cooperando para também considerar a rede como um todo, deverá ser capaz de minimizar o consumo de recursos e melhorar a produtividade e qualidade dos serviços providos. O capítulo a seguir descreve o projeto de simulação realizado nesse trabalho

com o intuito de demonstrar a viabilidade de implementação e instalação de um ESA em nós sensores, e os benefícios esperados para o mesmo.

Conforme mencionado em capítulos anteriores, o projeto de um modelo EA deve ser algo particular e específico para o tipo de rede, elementos de rede e aplicações desejadas. Nesse caso, o modelo de ESA apresentado acima não seria totalmente adequado para ser aplicado em redes locais sem fio, sistemas embutidos específicos ou quaisquer outros tipos de redes diferentes de RSSFs. Nesses casos, novos estudos sobre as particularidades de cada rede devem ser realizados e modificações certamente executadas sobre o modelo de ESA proposto.

Capítulo 5

Projeto de Simulação

Este capítulo apresenta a descrição detalhada de uma instância do modelo de ESA proposto nessa dissertação, considerando-se a plataforma de nós sensores sem fio Mica Motes 2. Essa instância foi desenvolvida com o objetivo de se demonstrar a viabilidade de implementação do modelo de EA proposto para RSSFs, bem como os benefícios que os mesmos podem trazer para os componentes dessas redes, em termos de uso eficiente de recursos, aumento de produtividade, precisão dos dados entregues ao observador e garantia da qualidade dos serviços providos. A Seção 5.1 descreve quais são os objetivos do processo de avaliação, as métricas consideradas e o processo de avaliação utilizado. A Seção 5.2 apresenta o funcionamento geral da instância proposta, identificando as funções utilizadas em cada serviço do laço autônomo. Além disso descreve as suposições consideradas para a RSSF simulada e seus componentes. A seção descreve ainda os detalhes de implementação das funcionalidades de classificação de dados, correlação de eventos e as políticas de reprogramação e reconfiguração utilizadas. A Seção 5.3 apresenta a ferramenta de simulação utilizada, apontando os motivos que levaram a escolha da mesma, os três diferentes cenários elaborados para se comparar o comportamento de nós sensores com e sem a instância do ESA embutida e também todos os valores utilizados para os parâmetros de simulação ligados as características da rede, elementos de rede, aplicação e da própria simulação. A Seção 5.4 apresenta as considerações finais para o capítulo. Experimentos realizados com o intuito de demonstrar a viabilidade de implementação de ESAs em nós sensores Mica Motes 2 e também realizar avaliação funcional da instância proposta, estão apresentados no Apêndice A.

5.1 Metodologia

Neste trabalho, além de propor o elemento autônomo para RSSFs (ESAs), foi criada uma instância do modelo proposto para a plataforma de nós sensores Mica Motes 2 [Hill e Culler, 2002]. Como caso de estudo, esse modelo foi instanciado considerando RSSFs planas, ou seja, sem formação de grupos, e homogêneas, com todos os nós possuindo as mesmas características de hardware (ver Seção 1.1).

Avaliações funcionais e de desempenho de nós sensores que possuem a instância de ESA

proposta embutida (RSSFs com ESA localizado) serão simuladas, comparando-as com avaliações feitas para nós que não possuem essa instância (RSSFs não-gerenciadas). Além disso, serão também comparados ESAs que realizam cooperação, trocando mensagens de gerenciamento entre si (RSSFs com ESA distribuído), com outros que executam suas ações de gerenciamento de forma localizada e individual.

Os objetivos de se realizar as avaliações descritas acima são: verificar se um ESA se comporta da forma esperada, realizando de forma correta cada funcionalidade do ciclo autônomo; Demonstrar a viabilidade de implementação desses elementos autônomos em nós sensores sem fio, apresentando o consumo de recursos para os mesmos; Apresentar as características de desempenho de nós sensores autônomos (com instância de ESA embutida) tais como produtividade, precisão e qualidade de serviço; Avaliar o impacto causado pelo uso de cooperação entre ESAs por meio da troca de mensagens de gerenciamento.

Como forma de alcançar os objetivos descritos acima, o seguinte conjunto de métricas a serem utilizadas para avaliação funcional e de desempenho foi definido:

- Consumo total de energia
- Porcentagem do consumo de energia com comunicação, processamento e sensoriamento
- Quantidade de bytes enviados pelos nós sensores considerando a aplicação e os serviços autônomos implementados
- Porcentagem de pacotes perdidos
- Número de pacotes enviados e perdido
- Número total de mensagens enviadas
- Número total de dados enviados
- Número de dados relevantes enviados
- Número total de mensagens recebidas
- Número de mensagens relevantes recebidas
- Número de dados relevantes recebidos
- Consumo de memória.

O processo de avaliação será realizado por meio de simulações. Apesar de não apresentar tantos detalhes como os experimentos reais, a avaliação por simulação pode prover um bom nível de detalhamento, dependendo da ferramenta escolhida para utilização. Além disso, esse tipo de avaliação permite maior liberdade para testes com grande número de elementos.

5.2 Instância do ESA para Plataforma Mica Motes 2

A instância do ESA contém implementações simples de cada um dos quatro serviços definidos para um ESA (ver Seção 4), além de uma base de conhecimento que possui MIB, AIB, máquina de políticas e CPB (Seção 4.6). A Figura 5.1 ilustra o funcionamento geral da instância de ESA projetada.

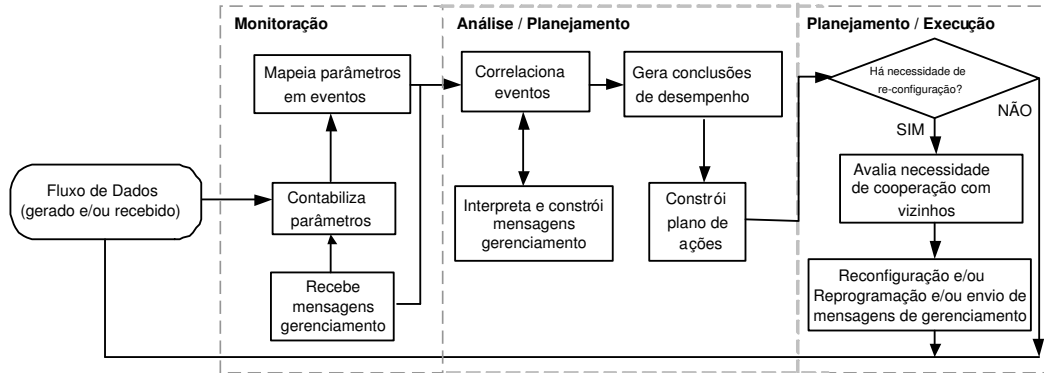


Figura 5.1: Instância do modelo de ESA implementada para as simulações deste trabalho.

O serviço de monitoração contabiliza os seguintes parâmetros de hardware e software: energia residual, número de dados relevantes coletados (dados que apresentem valores acima da média esperada), número de dados relevantes coletados pelos vizinhos, porcentagem de perda de mensagens e quantidade de bytes transmitidos. Também são monitorados os intervalos de sensoriamento e disseminação. A monitoração é periódica, sendo realizada a partir da análise do fluxo de dados gerados pelo próprio nó e/ou recebidos de seus vizinhos, e leitura de informações do hardware e software (auto-consciência e auto-conhecimento). Nos cenários que possuem RSSFs autônomicas com ESAs distribuídos, os ESAs de nós vizinhos são capazes de trocar mensagens de controle entre si (ver Figura 5.2(b)). Tais mensagens contêm dados contabilizados e de configuração e são trocadas periodicamente. Neste caso, mais um parâmetro pode ser monitorado: porcentagem de vizinhos com configurações semelhantes. Caso um nó perceba que suas configurações estão muito próximas a de seus vizinhos, ele pode decidir se retirar de serviço temporariamente. Essa mudança de estado administrativo requer reconfiguração de hardware, pois necessita que o nó modifique os estados de componentes, tais como rádio, processador e sensores. Todas as informações obtidas com este serviço são armazenadas na MIB e AIB do ESA.

O serviço de análise é executado em seguida. Nele, as informações obtidas são convertidas em eventos, utilizando valores de limites inferiores e superiores, previamente definidos e armazenados na MIB e AIB do ESA. O valor de cada parâmetro monitorado é classificado como alto, baixo ou médio de acordo com parâmetros bem definidos armazenados na BC e cada classificação representa um evento. Os eventos são utilizados em um algoritmo de correlação, o qual apresentará conclusões sobre o desempenho do nó sensor, ou seja, verificará se o nível de qualidade de serviço (*Quality of Service* (QoS)) desejado está ou não sendo provido

(auto-diagnóstico). Um algoritmo de correlação baseado em regras [Meira, 1997] foi utilizado devido a adequação do mesmo às restrições computacionais do ambiente de execução. As regras no formato SE-ENTÃO verificam a ocorrência conjunta de alguns eventos e apontam para a satisfação ou não do nível de QoS desejado (Seção 5.2.5). Os dados contidos nas mensagens de controle recebidas no caso das RSSFs autonômicas com ESAs distribuídos também são convertidos em eventos, participando da correlação e análise de desempenho.

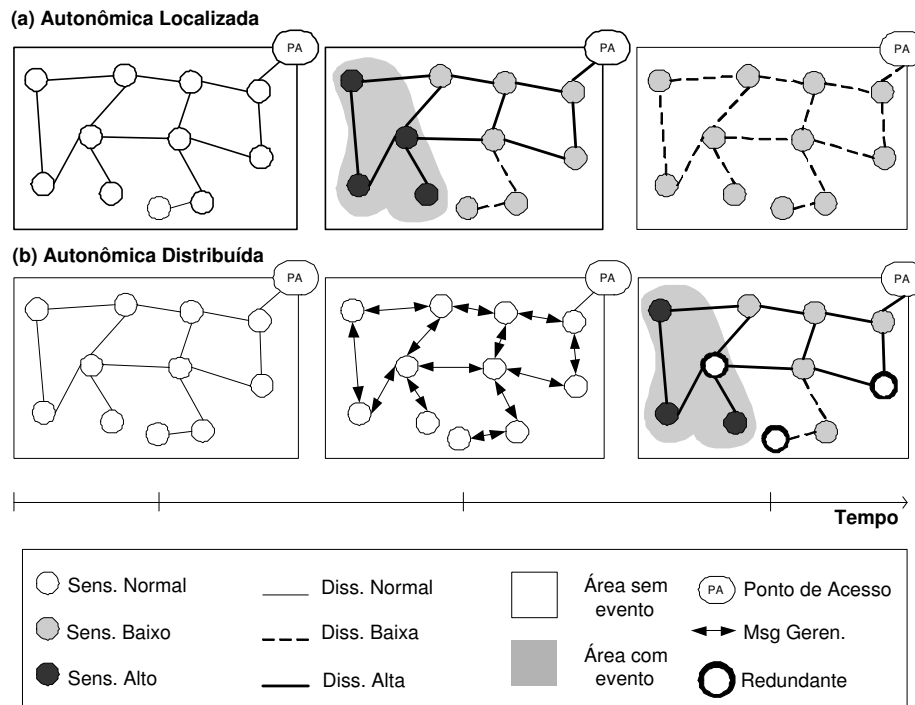


Figura 5.2: Ilustração do funcionamento das redes autonômicas localizada e distribuída projetadas.

De acordo com o resultado do serviço de análise, o serviço de planejamento deve verificar quais ações deverão ser tomadas. Para simplificar o projeto da instância de ESA utilizada neste trabalho, as opções de ações se restringiram ao aumento ou diminuição dos intervalos de sensoriamento e disseminação. Existe ainda a opção de mudança de estado administrativo (ativo ou inativo) para as RSSFs autonômicas com ESAs distribuídos. As mensagens de controle trocadas pelos nós desses cenários são planejadas por este serviço. As opções de ações disponíveis ficam armazenadas na MIB do ESA (auto-manutenção e auto-sustento). Finalmente, o serviço de execução recebe o plano de ações a serem executadas (incluindo o envio de mensagens) e o realiza de acordo com a melhor estratégia de execução possível (auto-configuração). Deve-se lembrar que reconfigurações de hardware e reprogramação de software (ou parâmetros) indicadas no plano de ações devem ser suportadas pela arquitetura do nó sensor utilizado. Todas as ações utilizadas neste trabalho podem ser efetivamente realizadas na plataforma Mica2 [Hill e Culler, 2002].

A Figura 5.2 ilustra um exemplo para a operação dos ESAs das redes autonômicas com

ESAs localizados e distribuídos. Os elementos da rede são iniciados com determinados valores para taxa de sensoriamento e comunicação considerados médios ou normais para a aplicação. Ao longo do tempo de vida da rede, eventos podem ocorrer em determinadas áreas da região monitorada. Nesse caso, os ESAs embutidos nos nós sensores na região do evento percebem o ambiente e, de acordo com suas configurações e disponibilidade de recursos, aumentam as taxas de coleta e envio dos nós sensores no intuito de elevar a precisão de entrega de informações para o PA. Os ESAs de nós que não estão presentes na área de um evento, mas que possuem a responsabilidade de roteá-los, executam ações para diminuição das taxas de sensoriamento e aumentam a taxa de disseminação. Dessa forma esses nós diminuem o envio de dados com pouca relevância sensorizados por eles e podem dar prioridade aos dados importantes enviados para o PA. Os ESAs de nós sensores que não estão em uma região de evento e não participam do roteamento de dados importantes, diminuem ambas as taxas de comunicação e sensoriamento desses nós. Nos momentos em que nenhum evento ocorre em parte alguma da região monitorada, os nós sensores permanecem em um estado onde apresentam pequena frequência de sensoriamento e disseminação.

A Figura 5.2(a) ilustra os ESAs localizados de uma rede autônoma aumentando sensoriamento e disseminação dos nós na região de evento, aumentando disseminação e diminuindo sensoriamento dos nós fora dessa região, mas responsáveis por rotear esses dados até o PA e diminuindo ambas as taxas para os nós que não coletam ou encaminham dados importantes. Os ESAs distribuídos das redes autônomas, ilustradas na Figura 5.2(b) possuem comportamento similar ao apresentado acima para a abordagem localizada. No entanto, existe uma etapa periódica de divulgação de dados, que permite a um ESA conhecer em detalhes as configurações atuais e dados monitorados de seus vizinhos, utilizando esse conhecimento para decidir retirar ou não o nó sensor de serviço temporariamente. Na figura, pode-se observar que os ESAs dos nós sensores apresentaram o mesmo comportamento daqueles da rede autônoma localizada, sendo que no entanto alguns nós foram retirados de serviços temporariamente após constatarem alto grau de similaridade com seus vizinhos.

Importante ressaltar que o mecanismo de inteligência computacional implementado pela instância proposta é o uso de intervalos inteligentes, ou seja, ajuste das taxas de sensoriamento e comunicação de acordo com a percepção que o ESA possui das características correntes do elemento gerenciado e de sua vizinhança.

5.2.1 Suposições Consideradas

A seguir estão descritas as suposições feitas sobre a rede e seus elementos para o projeto de simulação elaborado nesse trabalho:

- Os nós sensores que compõem a rede conhecem sua localização.
- A área de sensoriamento destes nós será considerada igual à de comunicação, sendo que ambas podem ser aproximadas por um círculo.

- Os elementos da rede são estáticos e possuem um relógio global aproximadamente sincronizado.
- A deposição dos nós da rede foi realizada seguindo uma distribuição uniforme.
- Por motivos de simplificação, considerou-se que os nós sensores não apresentarão falhas. Neste caso, eles funcionam normalmente até que atinjam o esgotamento de suas baterias.
- Os canais de comunicação não são confiáveis e pacotes podem ser perdidos e congestionamentos, bem como atrasos, podem ocorrer.
- Cada nó sensor considera como vizinhos aqueles que alcança através do roteamento utilizando apenas um passo.
- A aplicação considerada para a RSSF desenvolvida é de monitoração do nível de monóxido de carbono (CO) em um ambiente.
- Os serviços de auto-gerenciamento considerados pela instância de ESA projetada foram: auto-consciência, auto-conhecimento (serviço de monitoração), auto-diagnóstico (serviço de análise), auto-sustento, auto-manutenção (serviço de planejamento) e auto-configuração (serviço de execução). Esses serviços estão explicados em detalhes na Seção 2.4.

5.2.2 Requisitos de Memória

Dado que os nós sensores são dispositivos que possuem capacidade computacional limitada, o consumo de seus recursos é algo que deve ser realizado de forma racional e controlada. A memória é um desses recursos que deve ser estrategicamente utilizado. A decisão de se armazenar ou não determinados conteúdos (dados ou mesmo programas) deve ser realizada levando-se em consideração aspectos como frequência de utilização, prioridade, dentre outros. A utilização de memória em relação à instância do modelo de ESA proposto neste trabalho segue a seguinte formulação:

$$Total_{Memória} = Cod_{App} + Cod_{ESA} + BC \leq X \text{ KB}$$

onde Cod_{App} é a quantidade de bytes gasta pelo código da aplicação utilizada pela RSSF, Cod_{ESA} a quantidade necessária para o código contendo a implementação dos 4 serviços da instância do ESA ($Cod_{ESA} = Cod_{Monitoração} + Cod_{Análise} + Cod_{Planejamento} + Cod_{Execução}$), BC é o número de bytes para a Base de Conhecimento do ESA ($BC = MIB + AIB + CPB + Máquina \text{ Políticas}$) e $X \text{ KB}$ é o número de bytes que o nó sensor possui disponível em sua memória de programa. O código de uma aplicação simples desenvolvido para a plataforma Mica2 necessita de aproximadamente 20 KB. A quantidade de bytes para a memória de programa depende da arquitetura do nó sensor utilizada. O Mica2 possui 128 KB disponíveis para essa memória. Neste caso, Cod_{ESA} e BC poderiam utilizar no máximo aproximadamente 100 KB da memória de uma plataforma como a do Mica2.

5.2.3 Ferramenta de Simulação

As implementações deste trabalho foram realizadas utilizando-se o simulador NS-2 (*Network Simulator - 2*) [Network Simulator, 2006]. O NS-2 é um simulador discreto e dirigido a eventos bastante aceito e utilizado pela comunidade científica. Essa ferramenta já possui bibliotecas com a implementação de diversos protocolos, possibilitando a realização de simulações com diferentes tipos de redes, tais como redes tradicionais com fio, sem fio, *ad hoc* ou infra-estruturada, dentre outras. Em particular para as redes sem fio, já existem implementados importantes protocolos para camada de enlace e controle de acesso ao meio, tais como IEEE 802.15.4, também chamado ZigBee [ZigBee, 2006], e IEEE 802.11 [Wi-fi, 2006]. O alto nível de detalhamento do NS-2 e a estabilidade de suas implementações permitem que os resultados de simulação sejam confiáveis e próximos da realidade. Uma arcabouço para ferramenta NS-2 chamado Mannasim, proposto em [Braga et al., 2004, Lopes et al., 2006], possui um conjunto de classes base que permitem a simulação específica de RSSFs no NS-2. Com o uso do Mannasim é possível se desenvolver novas aplicações para RSSFs sem a necessidade de se preocupar com a implementar do comportamento básico dos nós sensores: sensoriamento de um ou mais parâmetros, processamento dos dados coletados e disseminação dos mesmos. As classes base disponíveis já possuem essas implementações e permitem inclusive a escolha de determinados aspectos, tais como tipos de sensoriamento, disseminação (contínuo, programado, dirigido a eventos e sob demanda), processamento e dispositivos sensores utilizados.

O simulador TOSSIM [Levis et al., 2003], específico para a plataformas que utilizam o sistema operacional TinyOS como os nós Mica Motes2, não foi utilizado por não possuir suporte para simulação de RSSFs com roteamento em múltiplos saltos. Além disso, esse simulador não apresentou característica de escalabilidade adequada para simulação com número de nós utilizado nesse trabalho.

5.2.4 Classificação de Dados

Um detalhe importante do projeto de simulação diz respeito aos limites e valores de incremento e decremento utilizados durante os ciclos de monitoração da instância de ESA projetada. Em particular, os serviços de análise e planejamento contam com limites e valores que devem ser definidos de acordo com a aplicação, o modelo computacional e o nível de detalhamento desejado para a simulação. Dentre estes estão valores para classificar eventos relacionados aos dados monitorados, valor de ajuste ou reconfiguração dos parâmetros de software reprogramáveis, e valores mínimos e máximos permitidos para estes parâmetros.

Em relação à classificação de eventos, os parâmetros monitorados ligados à quantidade de dados relevantes utilizam as seguintes porcentagens para classificação:

- Alto: $\geq 80\%$
- Médio: $> 20\%$ e $< 80\%$
- Baixo: $\leq 20\%$

Para os limites ligados aos intervalos, definiu-se que estes serão considerados:

- Altos (grandes) : 4 vezes maiores que o valor inicial
- Baixos (pequenos): 4 vezes menores que o valor inicial
- Médios: intervalo entre alto e baixo

Para a quantidade de bytes enviados:

- Alto: ≥ 15000 bytes
- Médio: intervalo entre alto e baixo
- Baixo: ≤ 500 bytes

Para a reconfiguração utilizamos como padrão o acréscimo ou decréscimo de 10 segundos nos intervalos até que os valores máximos ou mínimos sejam atingidos. Para o sensoriamento, foi permitido intervalo máximo de 120 segundos e mínimo de 10 segundos. Em relação à disseminação, máximo de 180 segundos e mínimo de 15 segundos.

As RSSFs autonômicas distribuídas contam ainda com um outro limite, utilizado para verificar o parâmetro de porcentagem de vizinhos com configurações semelhantes:

- Alto: $\geq 80\%$
- Baixo: $\leq 50\%$
- Médio: $> 50\%$ e $< 80\%$

Além disso, foi necessário utilizar dois limites que definem quando as informações de um certo nó estão ou não bastante semelhantes às de um vizinho. Neste trabalho, caso 60% dos dados de um nó estejam diferentes dos de um vizinho em apenas 20%, considera-se que as informações destes dois nós são similares. Os cenários 2 e 5 possuem RSSFs autonômicas localizadas cujos nós não são capazes de realizar mudança de estado administrativo.

5.2.5 Correlação de Eventos

Conforme mencionado anteriormente, o algoritmo utilizado nesse projeto de simulação foi a correlação baseada em regras. A seguir estão descritas as regras implementadas nos ESAs dos nós sensores das redes autonômicas (localizadas e distribuídas) simuladas. Vale ressaltar que a classificação de um dado em alto, médio ou baixo, conforme descrito na Seção 5.2.4 acima, é realizada utilizando-se limites fixos e bem definidos, armazenados na MIB ou AIB do ESA. As Figuras 5.3 e 5.4 contém fluxogramas que ilustram as regras de correlação utilizadas nas simulações.

- SE a porcentagem de dados relevantes for alta ou média e o número de bytes transmitidos alto e energia residual alta ou média e intervalo de sensoriamento maior do que o mínimo permitido, ENTÃO intervalo de sensoriamento inadequado (muito alto) e intervalo de disseminação adequado;

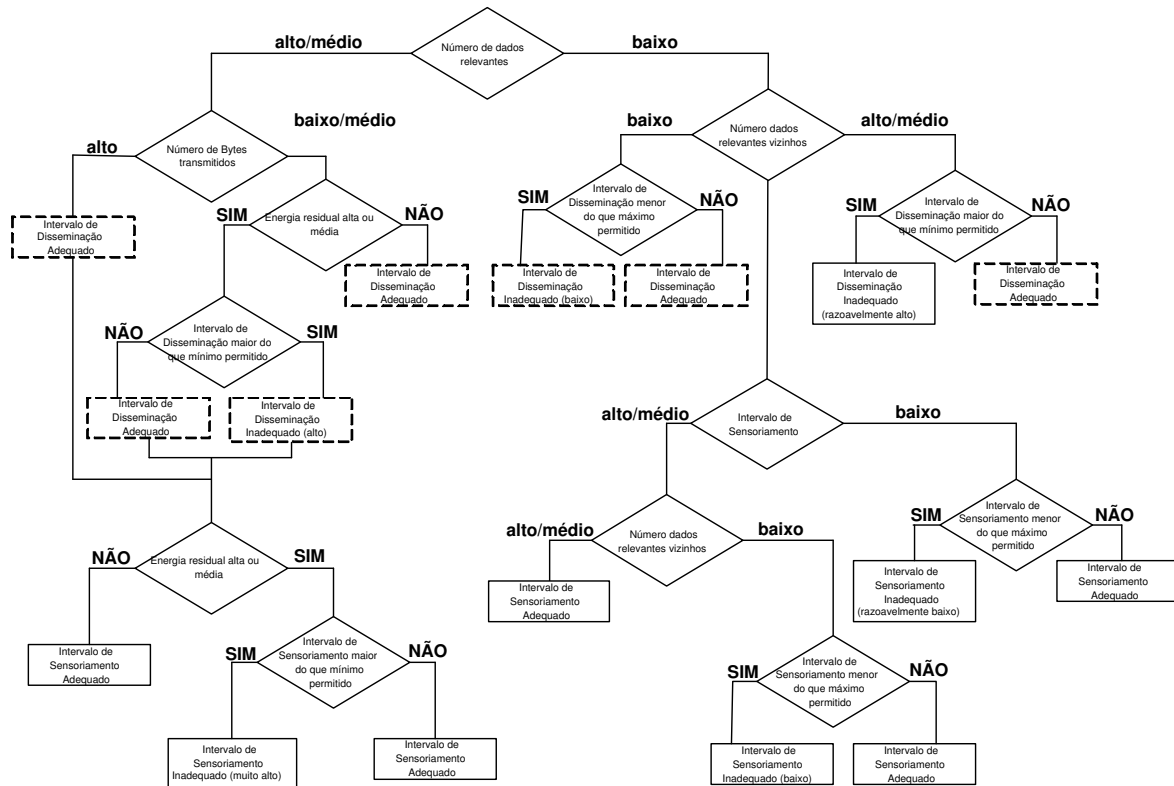


Figura 5.3: Fluxograma para regras de correlação relacionadas com reprogramação utilizadas nas simulações.

- SE a porcentagem de dados relevantes for alta ou média e o número de bytes transmitidos baixo ou médio e energia residual alta ou média e intervalo de disseminação e sensoriamento maiores que os mínimos permitidos, ENTÃO intervalos de sensoriamento e disseminação inadequados (altos);
- SE a porcentagem de dados relevantes for baixa, o intervalo de sensoriamento alto ou médio e a porcentagem de dados relevantes dos vizinhos baixa e intervalos de disseminação e sensoriamento menores do que os máximos permitidos, ENTÃO intervalos de sensoriamento e disseminação inadequados (baixos);
- SE a porcentagem de dados relevantes for baixa, o intervalo de sensoriamento alto ou médio e a porcentagem de dados relevantes dos vizinhos alta ou média e intervalo de disseminação maior do que o mínimo permitido, ENTÃO intervalo de sensoriamento adequado e intervalo de disseminação inadequado (razoavelmente alto);
- SE a porcentagem de dados relevantes for baixa, o intervalo de sensoriamento baixo e a porcentagem de dados relevantes dos vizinhos baixa e intervalos de sensoriamento e disseminação menores do que os máximos permitidos, ENTÃO intervalo de sensoriamento inadequado (razoavelmente baixo) e intervalo de disseminação inadequado (baixo);

- SE a porcentagem de dados relevantes for baixa, o intervalo de sensoriamento baixo e a porcentagem de dados relevantes dos vizinhos alta ou média e intervalo de sensoriamento menor do que o máximo permitido e intervalo de disseminação maior do que o mínimo permitido, ENTÃO intervalo de sensoriamento inadequado (razoavelmente baixo) e intervalo de disseminação inadequado (razoavelmente alto);
- SE a porcentagem de vizinhos com configurações semelhantes aos dados locais é alta, ENTÃO alta redundância;
- SE a porcentagem de vizinhos com configurações semelhantes aos dados locais é média, ENTÃO média redundância;
- SE a porcentagem de vizinhos com configurações semelhantes aos dados locais é baixa, ENTÃO baixa redundância.

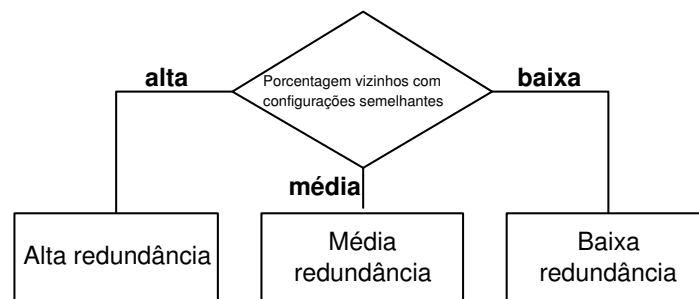


Figura 5.4: Fluxograma para regras de correlação relacionadas com reconfiguração de hardware utilizadas nas simulações.

5.2.6 Políticas para Reprogramação e Reconfiguração

Após a realização da correlação, o serviço de planejamento aplica políticas de acordo com as conclusões obtidas. A seguir estão descritas as políticas utilizadas nesse trabalho, as quais associam conclusões obtidas pelo serviço de análise a ações pré-determinadas de reprogramação ou reconfiguração:

- **Condição:** Intervalo (disseminação ou sensoriamento) inadequado (muito alto);
Ação: Diminuir intervalo;
- **Condição:** Intervalo (disseminação ou sensoriamento) inadequado (muito baixo);
Ação: Aumentar intervalo;
- **Condição:** Intervalo (disseminação ou sensoriamento) inadequado (alto);
Ação: Diminuir intervalo com probabilidade 80%;
- **Condição:** Intervalo (disseminação ou sensoriamento) inadequado (baixo);
Ação: Aumentar intervalo com probabilidade 80%;

- **Condição:**Intervalo (disseminação ou sensoriamento) inadequado (razoavelmente baixo);
Ação:Aumentar intervalo com probabilidade 20%;
- **Condição:**Intervalo (disseminação ou sensoriamento) inadequado (razoavelmente alto);
Ação:Diminuir intervalo com probabilidade 20%;
- **Condição:**Alta redundância;
Ação:Mudar para estado inativo com probabilidade 60%;
- **Condição:**Média redundância;
Ação:Mudar para estado inativo com probabilidade 40%;
- **Condição:**Baixa redundância;
Ação:Permanecer ativo.

5.2.7 Mensagens Utilizadas

As RSSFs simuladas nesse trabalho utilizam dois tipos de mensagem: mensagem para disseminação dos dados coletados e mensagens de gerenciamento para disseminação de informações de controle e configuração. O primeiro tipo de mensagem é utilizada tanto pelas redes autônomicas (distribuídas e localizadas) quanto pelas redes não-gerenciadas, para envio dos dados sensorizados para o observador. O segundo tipo é utilizado apenas pelas redes autônomicas com ESAs distribuídos para a troca de informações de gerenciamento entre os ESAs embutidos nos nós comuns, em intervalos regulares de tempo.

(a) Mensagem de dados

Tipo Msg	Tipo Evento	Id. Nó	Tempo	Dados Coletados
ALARME	SENSOR_REPORT	INT-16BITS	REAL-16BITS	VETOR INT-16BITS

(b) Mensagem de gerenciamento

Tipo Msg	Tipo Evento	Id. Nó	Tempo	Parâmetros Monitorados
ALARME	INFORM	INT-16BITS	REAL-16BITS	VETOR INT-16BITS

Figura 5.5: Mensagens utilizadas pelos ESAs das redes simuladas.

A Figura 5.5 ilustra o formato dos dois tipos de mensagens utilizadas, apresentado os campos presentes em cada uma delas. Pode-se observar que as mensagens de dados (Figura 5.5(a)) é formada por campos para tipo da mensagem, tipo de evento, identificador do nó sensor, momento de envio da mensagem e um vetor contendo dos dados coletados. Os quatro campos iniciais possuem 2 Bytes cada. O vetor de dados coletados possui tamanho variável, sendo cada posição do mesmo composta por 2 Bytes. A mensagem para troca de informações de gerenciamento (Figura 5.5(b)) é bastante similar aquela usada para troca de dados. Ela possui também os campos para tipo da mensagem, tipo de evento, identificador e tempo de

envio, cada um desses apresentando 2 Bytes. Em seguida, essa mensagem contém um vetor para as informações de controle e configuração a serem enviadas. As informações enviadas nesse vetor pelos ESAs distribuídos desse trabalho são: número de dados relevantes, número de dados relevantes de vizinhos, número de bytes transmitidos, energia residual, intervalo de sensoriamento, intervalo de comunicação. Cada campo desse vetor também possui 2 Bytes. Assim sendo, a mensagem de troca de informações de gerenciamento possui 20 Bytes no total.

5.3 Descrição dos Parâmetros de Simulação

Para realizar o roteamento nos cenários simulados, um algoritmo chamado EFTREE [Figueiredo et al., 2004] foi utilizado. Este algoritmo, que é específico para RSSFs planas, constrói uma árvore de roteamento com os nós da rede, a partir do PA da mesma. Em resumo, o PA, o qual será a raiz da árvore, envia uma mensagem de difusão (*broadcast*) para os nós vizinhos. Estes nós, ao receberem a mensagem, reconhecem o PA como pai, e a repassam para seus vizinhos. O processo se repete até que nós que não possuem vizinhos que ainda não receberam a mensagem sejam alcançados. Este algoritmo permite reconstrução periódica da árvore de roteamento. No entanto, como o modelo de falhas deste trabalho não inclui falhas dos nós sensores, a opção de reconstrução não foi utilizada.

Os nós sensores retirados de serviço temporariamente pela reconfiguração de hardware dos cenários autônomicos com ESAs distribuídos, continuam a realizar roteamento dos dados recebidos por seus vizinhos. Para verificar o comportamento dos nós mediante o aumento inesperado do valor médio do parâmetro sensoriado, a ocorrência de um evento durante a simulação foi planejada. Neste momento, alguns nós da rede passam a coletar dados com valores acima do esperado.

A aplicação escolhida como estudo de caso para a RSSF simulada foi a monitoração de monóxido de carbono (CO) em ambientes fechados e controlados. Uma monitoração desse tipo é importante por exemplo em galpões industriais de empresas de metalurgia ou petroquímica, visto que as atividades executadas nesses ambientes podem produzir quantidades relevantes desse tipo de gás como sub-produto de queimas, o qual em determinadas taxas é prejudicial para a saúde dos seres humanos.

Três tipos de cenários de simulação foram construídos, sendo dois deles compostos por RSSFs autônomicas, cujos nós possuem uma instância de um ESA embutido e um contendo uma RSSF não-gerenciada.

- **Cenário contendo ESA Localizado:** nós sensores possuem a instância do ESA apresentada na Seção 5.2 embutida. Nesse caso, gerentes autônomicos supervisionam, controlam e representam os nós sensores, executando o laço autônomico em intervalos regulares de tempo. O comportamento de uma rede desse tipo está ilustrada na Figura 5.2(a).
- **Cenário contendo ESA Distribuído:** nós sensores também possuem uma instância do ESA embutida, e são capazes de trocar mensagens de gerenciamento entre si,

realizando assim uma certa cooperação para realização de reserva de recursos, já que nesse caso podem executar a ação de se retirarem de serviço temporariamente, caso suas configurações estejam bastante similares as de seus vizinhos.

- **Cenário sem ESA:** neste tipo de cenário, os nós sensores não possuem ESA embutido. Neste caso, os elementos de rede apenas sensoriam e disseminam seus dados, utilizando sempre suas configurações iniciais.

Cada tipo de cenário descrito acima foi simulado para 100 e 200 nós. A Tabela 5.1 apresenta cada um dos cenários implementados. As características destes cenários (rede, nós sensores, eventos, dentre outros aspectos), descritas a seguir, foram definidas de acordo com a aplicação e o modelo computacional utilizados. O número máximo de nós sensores simulados foi mantido em 200 devido a restrições computacionais. Em média são necessárias 3 horas para execução de todos os cenários (Tabela 5.1) uma única vez. Todos os resultados foram obtidos com o uso de uma máquina de simulação com as seguintes características: processador Pentium 4 de 2,4GHz, 1GB de memória RAM e 2 discos rígidos de 80GB cada. Diferentes valores foram testados para os intervalos de monitoração e troca de mensagens de controle. Foram escolhidos os valores que melhor representavam o comportamento da rede (melhor custo-benefício).

Tabela 5.1: Cenários implementados. A densidade das redes utilizadas foi sempre mantida em 1 nó/400m².

Cenário	Sigla	Tipo	# Nós
1	ESA_{AD1}	ESA Autônomo Distribuído	100
2	ESA_{AL2}	ESA Autônomo Localizado	100
3	NG_3	Sem ESA (Não gerenciada)	100
4	ESA_{AD4}	ESA Autônomo Distribuído	200
5	ESA_{AL5}	ESA Autônomo Localizado	200
6	NG_6	Sem ESA (Não gerenciada)	200

- **Simulação:** Tempo de simulação: 10.000s; Número de simulações: 33; Intervalo de confiança: 95%.

- **Rede:** Número de nós: 100 ou 200; Organização: plana; Configuração: homogênea; Tamanho da Área (X e Y): 200m x 200m (para 100 nós) e 282m x 282m (para 200 nós); Densidade: 1 nó a cada 400m²; Localização PA: perímetro; Distribuição dos nós: uniforme.

- **Elementos de Rede:** Intervalo inicial de sensoriamento: 20s; Intervalo inicial de disseminação: 40s; Intervalo de monitoração (ESA): 100s; Intervalo de mensagens de gerenciamento: 200s; Intervalo de desligamento de redundantes: 300s; Consumo tx: 0.036W; Consumo rx: 0.024W; Consumo com sensoriamento: 0.0021W; Consumo com processamento: 0.024W; Largura de banda: 19.2 Kbps; Tipo de sensoriamento: programado; Tipo de disseminação:

programada; Pilha de protocolos: MannaNMP [Silva et al., 2005]/EFTREE/802.15.4; Mobilidade: não; Capacidade bateria: 1J; Alcance de sensoriamento e comunicação: 40m.

- **Aplicação:** Parâmetro sensoriado: monóxido de carbono (CO); Valor médio: $43mg/m^3$; Máximo valor permitido: $60mg/m^3$; Número de eventos: 1; Duração do evento: 1000s; Início do evento: 100s; Localização do evento: metade esquerda da área monitorada; Valor sensoriado durante evento: $80mg/m^3$.

Todas as características de consumo de energia utilizadas para os nós sensores, foram configuradas de acordo com valores reais apresentados pelos nós Mica2. Em relação ao alcance de comunicação, experimentos realizados com a plataforma Mica2 [SensorNet, 2006] demonstraram que em geral esses nós conseguem um alcance de transmissão entre 10 e 70 metros, de acordo com as condições do ambiente e posicionamento dos nós e PA. O valor de 40 metros escolhido para utilização está dentro das possibilidades reais de alcance desses nós. Os intervalos utilizados para execução do laço autonômico, troca de mensagens de gerenciamento e intervalo de desligamento de redundantes foram escolhidos de acordo com um conjunto de opções testadas. Os valores escolhidos foram aqueles que apresentaram melhores relações de custo-benefício, possibilitando bons resultados sem consumo exagerado de recursos. Os valores escolhidos para o parâmetro sensoriado (CO) estão de acordo com os níveis médios e máximos de presença desse gás na atmosfera aceitos por lei pelos municípios e cidades [Secretaria Municipal de Governo, 2006].

Para garantir a cobertura da área monitorada, conectividade e ainda realizar a reserva de recursos para prolongar o tempo de vida, uma RSSF densa foi projetada. Nesse caso, considerando-se o alcance de 40 metros e distribuição uniforme utilizada, destinou-se uma área de aproximadamente 20m x 20m para cada nó sensor, mantendo a densidade da rede estabelecida em 1 nó a cada $400m^2$. Assim, os valores utilizados para as dimensões X e Y da área monitorada são calculadas de acordo com o número de nós e a densidade estabelecida acima.

5.4 Considerações Finais

Este capítulo apresentou os detalhes referentes ao projeto de simulação construído para esse trabalho. O objetivo das simulações é a demonstração da viabilidade de desenvolvimento de uma instância do modelo de ESA proposto nessa dissertação em nós sensores, considerando-se a plataforma Mica Motes 2 em particular. Além disso, as simulações também deverão apontar as características de desempenho dos nós sensores que possuem a instância de ESA embutida e os benefícios que a mesma proporciona em termos de consumo de recursos, produtividade e qualidade de serviços.

Foram apresentados no capítulo a metodologia considerada para realização das simulações e a instância de ESA construída para a plataforma Mica2, detalhando-se o seu funcionamento através da descrição das ações executadas em cada um dos serviços do laço autonômico. As

funcionalidades de classificação de dados, correlação de eventos e políticas para reprogramação e reconfiguração implementadas foram discutidas em detalhes. Também foram descritas as suposições consideradas na construção dos cenários simulados, a ferramenta de simulação utilizada, tipos de mensagens utilizadas, e os requisitos de memória observados para se embutir um ESA em nós sensores Mica2.

A última seção do capítulo apresentou os tipos de cenários simulados, construídos para se chegar aos objetivos da simulação. Cada um dos cenários simulados específicos foi listado em uma tabela de cenários. Os parâmetros utilizados de forma específica para configuração da simulação, rede, elementos de rede e aplicação também foram descritos.

Capítulo 6

Análise de Resultados

Neste capítulo serão apresentados e analisados os resultados obtidos com a simulação da instância de um ESA criada para a plataforma de nós sensores Mica Motes2, conforme descrita no Capítulo 5. Os parâmetros de desempenho considerados nessa análise foram consumo de energia, quantidade de pacotes perdidos, precisão da informação enviada pelos nós sensores e recebida pelo PA e consumo de memória. Cada um desses parâmetros foram ainda avaliados considerando a variação do número de nós, da duração do intervalo de monitoração (intervalo de tempo entre execuções das iterações do ciclo autonômico), do número de eventos ocorridos e da quantidade de processamento exigida pelo ciclo autonômico. As Seções 6.1, 6.2, 6.3, 6.4 e 6.5 apresentam respectivamente os resultados encontrados para consumo de energia, precisão da informação enviada, precisão da informação recebida, perda de pacotes e consumo de memória. As Seções 6.6, 6.7 e 6.8 apresentam resultados de simulação para a variação do intervalo de monitoração, quantidade de eventos ocorridos na região monitorada e número de instruções necessárias para execução de uma iteração do ciclo autonômico do ESA. A Seção 6.9 apresenta uma conclusão geral sobre os resultados de simulação obtidos.

6.1 Consumo de Energia

Uma vez que as RSSFs possuem uma fonte limitada e finita de energia devido a características tais como dimensões reduzidas e operação desacompanhada em locais inóspitos ou hostis, é importante verificar o impacto causado pela solução de ESA proposta no consumo desse recurso. Os resultados observados para essa métrica, tanto em termos do consumo total quanto porcentagem de consumo por operação (comunicação, sensoriamento e processamento) estão apresentados a seguir.

A Tabela 6.1 mostra o valor médio em Joules, apresentados nas linhas brancas, e desvio padrão, apresentados nas linhas em cinza, do consumo total de energia dos nós comuns em cada um dos cenários. Pode-se observar que o consumo de energia dos nós nos cenários 1 e 4 (autonômicos distribuídos) é menor do que aquele dos demais cenários. As redes dos cenários 2 e 5, autonômicas localizadas, consumiram mais energia do que as dos cenários 1 e 4, uma vez que não eram capazes de fazer reconfigurações de hardware (desligar ou

modificar componentes dos nós sensores, para mudança de estado administrativo), mas ainda conseguiram apresentar um resultado melhor do que as redes não gerenciadas dos cenários 3 e 6. Os resultados apresentados foram obtidos considerando-se os parâmetros de simulação descritos no Capítulo 5.

Tabela 6.1: Consumo médio total de energia dos nós comuns em Joules e porcentagem média do consumo por serviço (sensoriamento, comunicação e processamento).

Cenário	1	2	3	4	5	6
	ESA_{AD1}	ESA_{AL2}	NG_3	ESA_{AD4}	ESA_{AL5}	NG_6
Consumo total (J)	0.61	0.93	0.97	0.61	0.97	0.99
	0.09	0.12	0.14	0.03	0.01	0.00
% Comunicação	98.00	84.56	93.70	98.09	87.77	94.86
	0.19	1.01	0.38	0.12	0.87	0.22
% Processamento	1.92	14.84	5.57	1.86	11.91	4.69
	0.17	1.00	0.33	0.12	0.86	0.19
% Sensoriamento	0.08	0.61	0.73	0.05	0.31	0.45
	0.02	0.13	0.11	0.01	0.05	0.07

Não houve um impacto significativo no consumo de energia devido ao aumento do número de nós da rede. Isso se deve principalmente ao fato de que a densidade da rede, para todos os cenários, foi mantida constante. A Tabela 6.1 mostra ainda a porcentagem média de consumo de energia dos nós comuns por serviço. Conforme esperado, em todos os cenários o consumo com comunicação é o mais significativo, seguido pelo processamento e finalmente sensoriamento. As redes autônomicas distribuídas apresentam o maior consumo com comunicação, devido às mensagens de gerenciamento trocadas. No entanto, tais redes apresentam um consumo médio aproximadamente 40% menor do que aquele dos demais cenários. Os cenários 2 e 5 consumiram menos com comunicação, pois enviaram menos mensagens de dados do que os cenários 3 e 6 e seus nós não realizaram troca de mensagens de gerenciamento.

6.2 Informações Enviadas pelos Nós Comuns

A Tabela 6.2 mostra nas linhas brancas os valores médios para quantidade total de mensagens, dados e dados relevantes enviados pelos nós de cada cenário durante o tempo de simulação. Os valores de desvio padrão para cada um desses itens são apresentados nas linhas em cinza. Considerando tanto os cenários contendo 100 nós, quanto os que contém 200, as redes autônomicas conseguem enviar um número total de mensagens menor por aumentarem seus intervalos de sensoriamento e disseminação em momentos adequados. No entanto, quando a análise do fluxo detecta a presença de um evento, tais intervalos são diminuídos, o que faz com que um grande número de amostras (dados) sejam enviados ao observador.

As RSSFs autônomicas distribuídas (cenários 1 e 4) enviaram um número menor de mensagens e dados do que as redes autônomicas localizadas, uma vez que durante o tempo de vida da rede existia sempre um conjunto de nós fora de serviço temporariamente. As redes

Tabela 6.2: Número de mensagens e dados enviados. Uma mensagem pode conter um ou mais dados sensoriados. Um dado relevante é aquele coletado durante a ocorrência de um evento.

Cenário	1	2	3	4	5	6
	ESA_{AD1}	ESA_{AL2}	NG_3	ESA_{AD4}	ESA_{AL5}	NG_6
Mensagens	5163.34	6637.06	8730.58	8070.38	10191.79	14310.74
	262.78	374.71	545.76	239.91	536.27	820.43
Dados	8255.22	10529.62	17361.21	13381.71	16465.97	28421.74
	396.42	998.69	1091.51	413.59	787.21	1640.88
Dados relevantes	3159.24	4378.35	2478.71	5762.35	7237.03	4701.53
	427.21	398.99	281.42	421.88	525.12	341.63

não gerenciadas enviaram sempre um número maior de mensagens, mas um número menor de dados relevantes. Os cenários 2 e 5 apresentaram resultados interessantes para esta métrica, visto que seus nós enviaram um número maior de dados relevantes do que os nós dos demais cenários. Vale ressaltar que os nós destes cenários (2 e 5) conseguem enviar mais dados relevantes, enviando menos mensagens do que os nós das redes não gerenciadas (cenários 3 e 6).

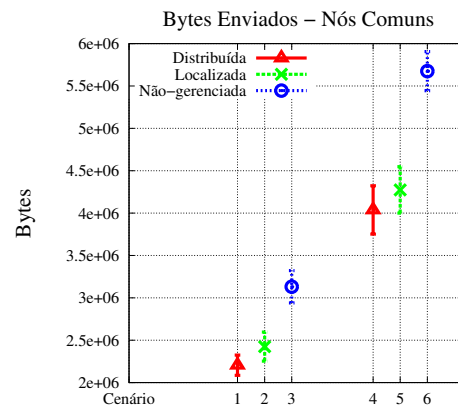


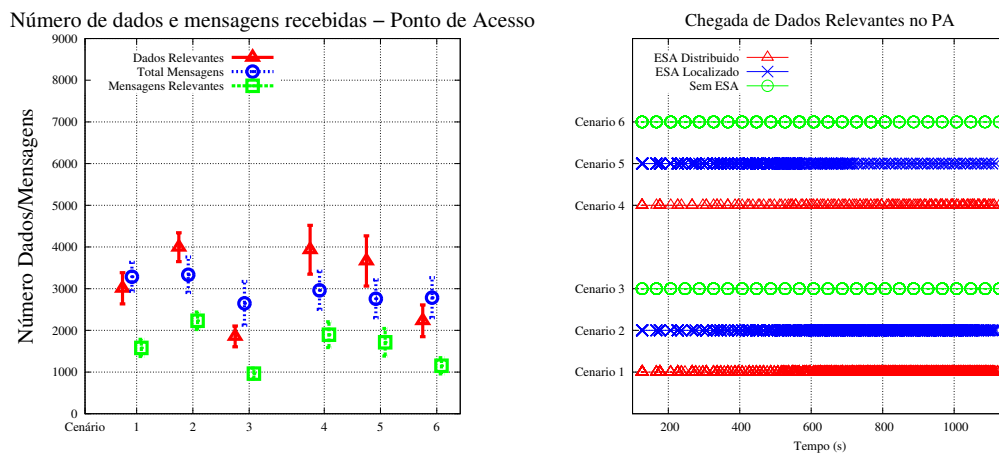
Figura 6.1: Médias e desvios padrão do total de Bytes enviados pelos nós comuns.

As redes autônomicas enviam grande número de dados e mensagens apenas nos momentos em que eventos ocorrem. No entanto, nos momentos em que nada ocorre no ambiente monitorado, os nós enviam um número menor de mensagens, as quais contém poucos dados sensoriados. Nenhum dado coletado é filtrado ou selecionado. O que muda nas redes autônomicas são os valores das taxas de sensoriamento e disseminação, que são ajustadas de acordo com a aplicação e com as condições internas e externas detectadas pelo ESA. A Figura 6.1 mostra que ao final, a quantidade de bytes enviados pelos nós dos cenários autônomicos é menor do que a dos cenários das redes não gerenciadas. Em particular, as redes distribuí-

das apresentaram um menor número de bytes enviados, devido à utilização da técnica de reconfiguração de hardware para realização de reserva de recursos.

6.3 Informações Recebidas pelo PA

As mensagens e dados produzidos e enviados pelos nós serão roteados pela rede até o PA. Perda de pacotes devido à colisões e congestionamento ocorrem devido à características do meio de transmissão. O gerenciamento deve implementar estratégias de utilização racional e produtiva do meio, de forma a maximizar a entrega de dados, em particular nos momentos em que eventos ocorrem no ambiente monitorado.



(a) Médias e desvios padrão do número de dados e mensagens recebidas pelo PA. (b) Chegada de dados relevantes no PA.

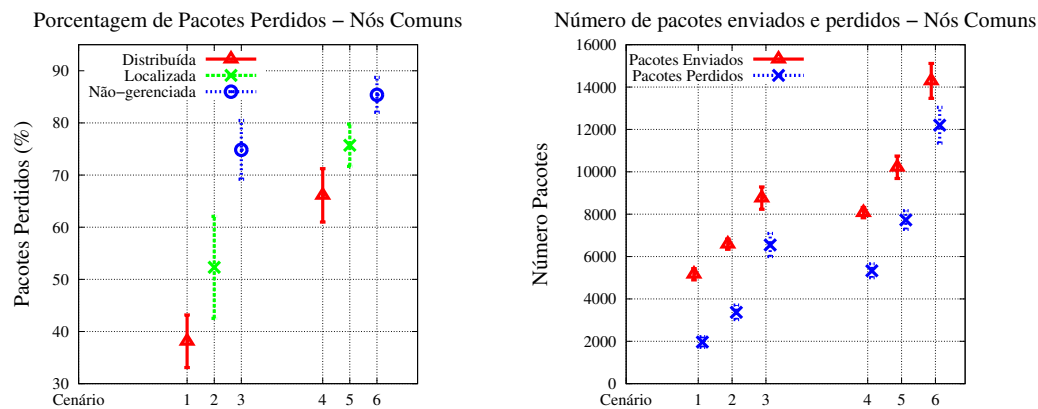
Figura 6.2: Avaliação da produtividade na chegada de mensagens e dados relevantes no PA.

O gráfico da Figura 6.2(a) mostra o número de dados, mensagens e mensagens relevantes recebidos pelo PA. A grande quantidade de dados e mensagens enviadas pelos nós sempre ativos da rede autônoma e localizada do cenário 2 não sobrecarregou o meio de comunicação, e permitiu uma maior entrega, principalmente de informações relevantes. No entanto, no cenário 5, o número alto de mensagens fez com que colisões e perdas ocorressem, e neste caso o cenário distribuído apresentou resultados ligeiramente melhores, apesar de ter enviado menor quantidade de dados e mensagens. Em geral, as redes autônomas obtiveram sempre melhor resultado do que aquelas não gerenciadas, devido à utilização estratégica dos recursos dos nós e da rede como um todo. A Figura 6.2(b) apresenta os momentos em que dados relevantes chegam ao PA quando ocorre um evento na área monitorada. Para construção do gráfico foi considerada uma amostra da chegada de dados no PA dentre as 33 obtidas com esse mesmo número de simulações. Pode-se observar que, enquanto nas redes sem ESAs os dados chegam em uma frequência constante, nas redes com ESAs localizados ou distribuídos, a chegada ocorre com maior intensidade. Uma vez que as redes com ESAs localizados não

retiram nós de serviço temporariamente, a chegada de dados nos cenários 2 e 5, no momento da ocorrência do evento, é ainda mais intensa. Isso se deve ao ajuste dos intervalos de sensoriamento e disseminação realizado pelo ESA nos momentos em que um evento ocorre na região monitorada. O aumento de produtividade e precisão na entrega das informações nos momentos mais apropriados observado através da atuação dos ESAs fica portanto ilustrado através desse gráfico.

6.4 Número de Pacotes Perdidos

A implementação dos serviços de gerenciamento também causa impacto sobre uma outra métrica importante: o número de pacotes perdidos. A utilização de técnicas de reconfiguração de hardware e reprogramação de software influenciam de forma positiva nesta métrica, conforme mostra o gráfico da Figura 6.3(a). A Figura 6.3(b) mostra a relação entre o número de pacotes enviados e o número de pacotes perdidos em cada um dos cenários. Os cenários autônômicos enviam um número menor de pacotes, o que se reflete em menor perda, consumo de energia e número de bytes enviados.



(a) Porcentagem de pacotes perdidos.

(b) Número de pacotes enviados e descartados.

Figura 6.3: Média e desvio padrão relativos às métricas de envio e perda de pacotes.

As redes não gerenciadas dos cenários 3 e 6 apresentaram os piores resultados. O grande número de pacotes enviados continuamente possivelmente levaram ao congestionamento do meio de transmissão, e conseqüentemente colisão e perda. O cenário 1 possui porcentagem de perda menor do que os cenários 2 e 3, assim como o cenário 4 em relação aos cenários 5 e 6. Ao relacionarmos a métrica de mensagens e dados enviados à porcentagem de pacotes perdidos, pode-se perceber que apesar das redes autônômicas localizadas enviarem um número maior de mensagens e dados relevantes (ver Seção 6.2), a perda de pacotes nestes cenários é maior do que aquela nos cenários 1 e 4, indicando sobrecarga no meio de transmissão. O impacto desta sobrecarga pode ser visto no gráfico da Figura 6.2(a), Seção 6.3. Para os cenários com

200 nós sensores, o PA recebeu menos dados e mensagens relevantes do cenário 5 do que do cenário 4, apesar do primeiro ter enviado quantidades maiores do que o último.

Em RSSFs existe um compromisso entre consumo de energia e precisão da informação. Quanto maior o número de dados coletados entregues ao observador, maior a precisão e conseqüentemente maior o consumo com sensoriamento, processamento e comunicação. Neste caso, é interessante que a rede controle o consumo de seus recursos em períodos em que eventos não ocorram no ambiente monitorado para poder consumi-los com maior intensidade nos momentos oportunos. As métricas avaliadas acima mostram como as redes autônomicas são eficientes em relação a esse compromisso, conseguindo minimizar o consumo de energia e ainda enviar uma maior quantidade de informações para o PA nos momentos adequados.

6.5 Consumo de Memória

A memória é um recurso computacional também bastante restrito em nós sensores, e portanto deve ser utilizado de forma racional. Na memória de um nó sensor devem ser armazenados o sistema operacional utilizado, contendo a aplicação para qual a rede foi construída, e também os dados coletados durante o tempo de vida deste. A solução de gerenciamento utilizada, implementada através da instância de ESA desenvolvida, deve estar integrada ao sistema operacional e a aplicação, não podendo portanto superar a quantidade de memória disponível (memória de programa), conforme descrito em 5.2.1.

Para verificar a quantidade de memória relativa à instância de ESA elaborada nesse trabalho, a mesma foi implementada em conjunto com uma aplicação de coleta dados de luminosidade disponível no sistema operacional TinyOS [TinyOS, 2006]. A aplicação em questão é chamada de *Surge* e, de acordo com a Tabela 6.3, consome aproximadamente 16KB de memória *Ready-Only Memory* (ROM) e 2KB de memória RAM (integrada ao sistema operacional). Nela, os nós sensores realizam coleta e disseminação contínua de dados de luminosidade, podendo realizar entrega para o PA em múltiplos saltos, por meio do algoritmo de roteamento utilizado. Além disso, os nós também estão preparados para receber mensagens vindas do PA com o objetivo de realizar mudanças em determinados parâmetros.

Tabela 6.3: Quantidade de memória RAM e ROM (em Bytes) necessária para cada um dos serviços (Monitoração, Análise, Planejamento e Execução) em conjunto com a BC, e para o ciclo autônomico em geral.

Serviço	Memória ROM (Bytes)	Memória RAM (Bytes)
Aplicação	15960	1847
Monitoração + BC	276	68
Análise + BC	352	75
Planejamento + BC	228	77
Execução + BC	170	70
Total Ciclo	1026	290
Total Geral	16986	2137

O ESA foi integrado à aplicação *Surge*. Dessa forma, além de exercer as tarefas dessa aplicação, os nós sensores podem executar em intervalos regulares de tempo (intervalo de monitoração) o ciclo autonômico (monitoração, análise, planejamento e execução). Cada uma das etapas do ciclo foi implementada em separado, com o intuito de se verificar quais eram seus impactos individuais. O serviço que apresentou maior consumo de memória ROM foi a análise. Isso se deve basicamente ao fato de que esse serviço é responsável pela implementação dos algoritmos de classificação de dados e correlação de eventos, sendo portanto o maior e mais complexo serviço da instância desenvolvida. O serviço de execução que apenas recupera o plano de ações da BC, monta e envia mensagens de gerenciamento, necessitou da menor quantidade de memória ROM, dentre os quatro serviços observados. Em termos de memória RAM, todos os serviços apresentaram aproximadamente a mesma necessidade de memória. No total, o ciclo autonômico representou um acréscimo de apenas aproximadamente 1KB para memória ROM e 290 Bytes para memória RAM. Esse resultado, em conjunto com a avaliação do consumo médio de energia (Seção 6.1), demonstra a viabilidade de implementação do ESA e indica a possibilidade de desenvolvimento de instâncias mais elaboradas, contendo inclusive os algoritmos de aprendizado de máquina e opções de classificação de dados e correlação de eventos mais avançadas, lembrando que os nós sensores Mica Motes 2 possuem 128KB disponíveis para a memória de programa (Seção 1.2.1).

6.6 Variação do Intervalo de Monitoração

O intervalo de monitoração é o tempo entre as execuções de iterações subseqüentes do ciclo autonômico. Um conjunto de simulações foi realizada com o objetivo de verificar o impacto causado pela modificação do intervalo de monitoração utilizado, isto é, observar o comportamento das métricas consideradas para intervalos menores ou maiores do que aquele utilizado para as simulações descritas nas seções anteriores. Nesse caso, todos os parâmetros de simulação foram mantidos iguais aos das simulações iniciais (ver Capítulo 5), modificando-se apenas o valor do intervalo de monitoração. O número de nós sensores utilizados foi mantido em 200. Não foi considerado o cenário para RSSFs não gerenciadas, visto que essas não utilizam o intervalo de monitoração, que é próprio dos ESAs.

Consumo de Energia: a Tabela 6.4 apresenta o consumo médio total e por serviço (comunicação, processamento e sensoriamento) de energia apresentado pelos nós sensores cujos ESAs executam ciclo de monitoração de 50 em 50 segundos, 100 em 100 segundos e 200 em 200 segundos. As linhas brancas apresentam os valores médios obtidos e as linhas em cinza os respectivos desvios padrão. Pode-se observar que, quanto menor o intervalo de monitoração, menor o consumo dos nós com comunicação e sensoriamento e maior o consumo com processamento. Cada execução de uma iteração do ciclo autonômico implica no processamento do conjunto de instruções associadas ao mesmo, e portanto, em um determinado consumo de energia. Quanto mais iterações, mais vezes serão executadas as instruções de um ciclo e portanto maior o consumo de energia. Com um intervalo de 50 segundos, os ESAs processam

o ciclo autônomo 200 vezes durante o tempo de simulação de 10.000 segundos, ou seja, 2 vezes mais do que os ESAs cujo intervalo é de 100 segundos e 4 vezes mais do que os ESAs com intervalo de 200 segundos. Assim, conforme o esperado, o consumo de energia médio com serviço de processamento é maior para os ESAs com menor intervalo.

Tabela 6.4: Consumo médio total de energia dos nós comuns em Joules e porcentagem média do consumo por serviço para os cenários 4 (autônomo distribuído) e 5 (autônomo localizado), contendo 200 nós e intervalo de monitoração variando entre 50, 100 e 200 segundos.

	Cenário 4 ESA_{AD4}			Cenário 5 ESA_{AL5}		
	50	100	200	50	100	200
Duração Intervalo (s)						
Consumo total (J)	0.59	0.61	0.71	0.99	0.95	0.97
	0.05	0.08	0.03	0.00	0.13	0.01
% Comunicação	97.40	98.11	98.42	79.91	88.04	93.49
	0.15	0.10	0.11	1.17	0.66	0.41
% Processamento	2.56	1.84	1.50	19.80	11.65	6.15
	0.14	0.10	0.10	1.19	0.66	0.40
% Sensoriamento	0.04	0.05	0.08	0.29	0.31	0.36
	0.01	0.01	0.01	0.05	0.06	0.06

Com a utilização de um valor baixo, como 50 segundos no contexto dessas simulações, o intervalo de monitoração se torna rapidamente pequeno em relação aos intervalos de sensoriamento e disseminação, principalmente nos momentos em que eventos não ocorrem. Os nós nesse caso passam a não ter a chance de realizar coleta e comunicação nos intervalos de execução do ciclo autônomo, chegando de forma bastante rápida ao limites máximos de intervalo para esses serviços. Assim, quanto menor a taxa de monitoração, menor a quantidade de comunicação e sensoriamento e portanto menor o consumo médio com esses dois serviços. Em relação ao consumo médio total de energia, para os ESAs do cenário 4 (autônomo distribuído), quanto menor o intervalo menor o consumo, devido ao fato de que os nós aumentam gasto com serviço de processamento enquanto diminuem gasto com serviço de comunicação, sendo que o primeiro tipo de serviço demanda menor quantidade de energia do que o segundo. Para os ESAs do cenário 5 (autônoma localizada), houve um consumo quase total da quantidade de energia disponível na bateria (1 Joule). Em todos os casos, o consumo foi bastante similar, sendo que o cenário com taxa de monitoração de 100 segundos apresentou consumo ligeiramente menor.

Mensagens, Dados e Bytes Enviados para o PA: sempre que o valor das taxas de sensoriamento ou disseminação é alterado, a contagem de tempo para que as mesmas expirem é reinicializada. Dessa forma, os nós possuem intervalos entre as execuções do ciclo menores do que aqueles para coleta e disseminação, fazendo com que em muitos casos esses serviços não sejam executados entre uma interação e outra do ciclo. Dessa forma, quanto menor o intervalo de monitoração, menor a quantidade de mensagens, dados e dados relevantes, conforme mostra a Tabela 6.5.

Tabela 6.5: Número de mensagens e dados enviados. Resultados obtidos para os cenários 4 (autonômica distribuída) e 5 (autonômica localizada) contendo 200 nós e variando intervalo de monitoração (50, 100 e 200 segundos).

Duração Intervalo (s)	Cenário 4 <i>ESA_{AD4}</i>			Cenário 5 <i>ESA_{AL5}</i>		
	50	100	200	50	100	200
Mensagens	6782.29	8045.97	9962.03	9202.70	10100.47	11083.47
	210.85	282.86	352.06	282.14	408.47	430.51
Dados	10950.00	13394.68	16917.09	14367.58	16313.66	18767.09
	294.90	410.13	506.53	449.13	637.09	594.83
Dados relevantes	5116.06	5893.23	6716.58	7224.30	7359.12	7771.78
	241.53	366.62	399.58	491.10	724.52	522.65

A Figura 6.4 ilustra a quantidade de bytes enviados pelos nós sensores durante o período de simulação, considerando-se a variação do intervalo de monitoração. Em conformidade com o exposto acima para justificação do consumo de energia e quantidade de dados e mensagens enviadas, as RSSFs com menores intervalos de monitoração enviam menor quantidade de bytes, visto que seus nós sensores geram menor quantidade de mensagens e dados.

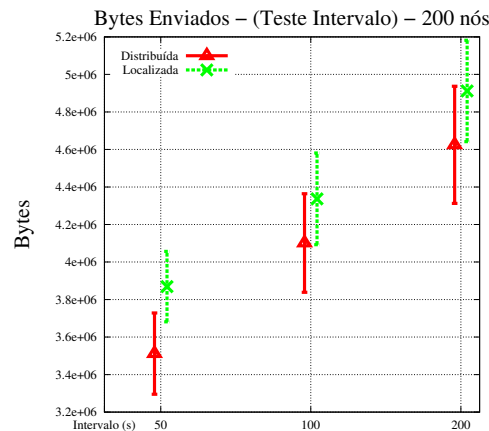
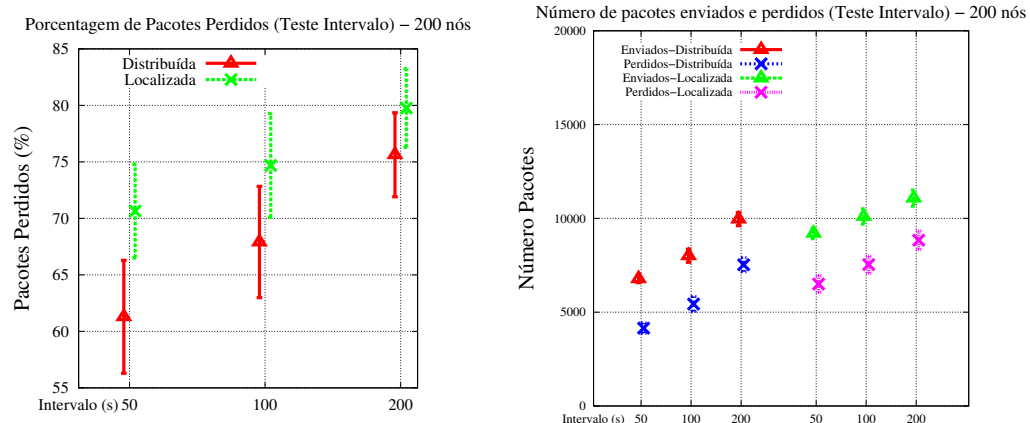


Figura 6.4: Médias e desvios padrão do total de bytes enviados pelos nós comuns. Resultados obtidos para os cenários 4 (autonômica distribuída) e 5 (autonômica localizada) contendo 200 nós sensores e variando a duração do intervalo de monitoração (50, 100 e 200 segundos).

Perda de pacotes: uma vez que as RSSFs com menores intervalos de monitoração acabam enviando uma menor quantidade de dados, a sobrecarga no meio de comunicação observada também é menor, o que reflete na menor porcentagem de perda de dados apresentada pelo gráfico da Figura 6.5(a). Em média, a diferença de perda de mensagens entre as RSSFs com 50 e 200 segundos de intervalo é de aproximadamente 15% para os ESAs distribuídos e 10% para os ESAs localizados. A Figura 6.5(b) mostra que, para ambos os cenários (localizado e

distribuído), as redes com menor intervalo de monitoração enviaram e perderam um número menor de pacotes.



(a) Porcentagem de pacotes perdidos.

(b) Número de pacotes enviados e descartados.

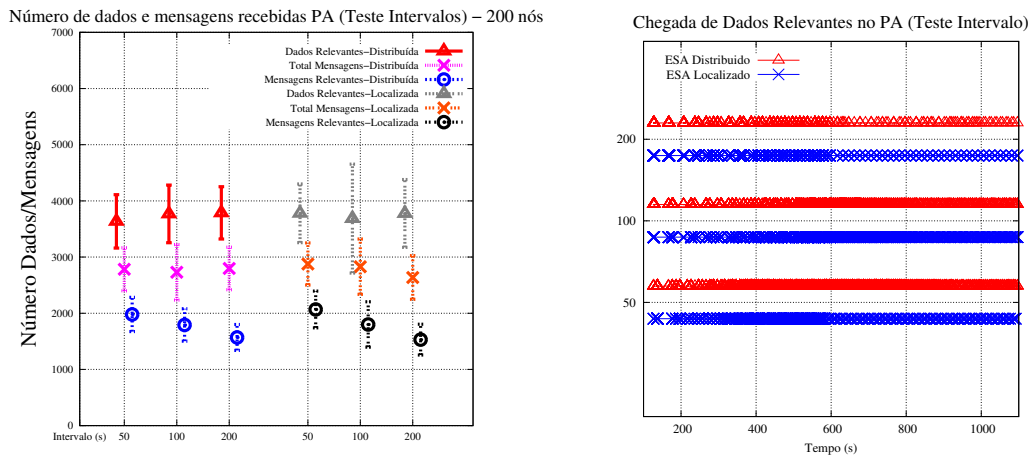
Figura 6.5: Média e desvio padrão relativos às métricas de envio e perda de pacotes para os cenários 4 (autônômica distribuída), 5 (autônômica localizada) e 6 (não gerenciada).

Dados e Mensagens recebidas pelo PA: em geral, conforme mostra a tabela 6.5, as RSSFs com ESAs localizados enviaram entre 1000 e 2000 dados relevantes a mais que aquelas com ESA distribuídos. No entanto, como a perda de pacotes para as primeiras foi maior de que para as segundas (Figura 6.5(a)), elas acabaram entregando basicamente a mesma quantidade de dados (Figura 6.6(a)). Esse mesmo argumento é válido para a entrega de mensagens e mensagens relevantes, e para a semelhança entre os valores observados para os diferentes valores de taxas de monitoração. Apesar das redes com menores taxas terem produzido menor quantidade de dados e mensagens, a menor taxa de perda permitiu que as mesmas acabassem entregando aproximadamente a mesma quantidade de dados e mensagens do que as demais. A Figura 6.6(b) ilustra um cenário para chegada de dados relevantes no PA.

Vale ressaltar que as relações entre os cenários 4 e 5, comentadas em detalhes nas Seções 6.1, 6.2, 6.3 e 6.4, se mantiveram para os resultados apresentados nessa seção.

6.7 Variação do Número de Eventos

Uma vez que os ESAs devem adaptar de forma autônômica as configurações dos nós sensores de acordo com a presença e ausência de eventos ocorridos no ambiente monitorado, foi realizado um conjunto de simulações variando-se a quantidade de eventos. Todas as métricas consideradas nesse trabalho foram coletadas e o objetivo é avaliar o funcionamento e desempenho dos ESAs localizados e distribuídos, observando o comportamento e a adaptação dos



(a) Médias e desvios padrão do número de dados e mensagens recebidas pelo PA. (b) Chegada de dados relevantes no PA.

Figura 6.6: Médias e desvios padrão do número de dados e mensagens recebidas pelo PA e produtividade na recepção de dados. Resultados obtidos para os cenários 4 (autônoma distribuída) e 5 (autônoma localizada) contendo 200 nós sensores e variando a duração do intervalo de monitoração (50, 100 e 200 segundos).

mesmos. As RSSFs sem ESAs (não gerenciadas) não foram incluídas nesses experimentos, visto que o comportamento dos nós sensores não se altera com a ocorrência ou não de eventos na região. Desse modo, o consumo de energia, bem como a precisão dos dados enviados e recebidos e a perda de pacotes, seriam os mesmos obtidos para as simulações apresentadas nas Seções 6.1, 6.2, 6.3 e 6.4. Foram consideradas a ocorrência de 1, 2 e 4 eventos, de acordo com os seguintes intervalos de tempo, todos com 500 segundos de duração:

1. **Evento 1:** 100 a 600 segundos;
2. **Evento 2:** 700 a 1200 segundos;
3. **Evento 3:** 1300 a 1800 segundos;
4. **Evento 4:** 1900 a 2400 segundos.

A configuração das simulações é a mesma apresentada pelo Capítulo 5, sendo modificadas apenas a quantidade do número de eventos e a duração dos mesmos, passando de 1000 para 500 segundos. Essa modificação foi realizada para evitar que o desgaste da bateria dos nós interferisse nos resultados de avaliação funcional e de desempenho da adaptação dos ESAs frente a ocorrência de um ou mais eventos.

Consumo de Energia: a Tabela 6.6 apresenta valores médios (linhas brancas) e desvios padrão (linhas em cinza) para o consumo médio total e por operação de energia, considerando-se a variação do número de eventos ocorridos na rede. Pode-se observar que, em todos os casos

apresentados pela tabela, o consumo total de energia permaneceu constante, independente do número de eventos ocorridos na região monitorada. Esses resultados se devem ao comportamento adaptativo dos ESAs, tanto das redes localizadas quanto distribuídas, sendo os mesmos capazes de coletar e disseminar com intensidade durante os períodos de eventos, mas diminuindo as taxas de sensoriamento e comunicação nos períodos em que nada ocorre na região monitorada.

Tabela 6.6: Consumo médio total de energia dos nós comuns em Joules e porcentagem média do consumo por serviço para os cenários 4 (autônomo distribuído) e 5 (autônomo localizado), contendo 200 nós e variando quantidade de eventos (1, 2 e 4).

	Cenário 4 <i>ESA_{AD4}</i>			Cenário 5 <i>ESA_{AL5}</i>		
	1	2	4	1	2	4
Número Eventos						
Consumo total (J)	0.60	0.62	0.61	0.97	0.97	0.98
	0.03	0.03	0.03	0.01	0.01	0.01
% Comunicação	98.12	98.11	98.01	86.88	88.09	88.75
	0.34	0.10	0.12	0.56	0.65	0.76
% Processamento	1.78	1.83	1.93	12.82	11.60	10.92
	0.10	0.09	0.12	0.56	0.65	0.76
% Sensoriamento	0.04	0.05	0.06	0.30	0.31	0.34
	0.01	0.01	0.01	0.05	0.05	0.06

Mensagens, Dados e Bytes Enviados para o PA: a Tabela 6.7 apresenta valores médios (linhas brancas) e respectivos desvios padrão (linhas em cinza) para a quantidade de mensagens, dados e dados relevantes enviadas pelos nós sensores. Conforme esperado, para ambos os cenários (localizado e distribuído), houve um aumento do número de mensagens, dados e dados relevantes enviados quando número de eventos na região monitorada também aumentou. Com a ocorrência de 4 eventos na região monitorada, os nós sensores enviaram em média mais do que o dobro de dados relevantes do que na ocorrência de apenas 1 evento.

Os valores apresentados pelas Tabelas 6.6 e 6.7 mostram a eficiência do comportamento adaptativo dos ESAs implementados: os nós sensores foram capazes de enviar uma grande quantidade de dados a mais nos cenários em que um número maior de eventos ocorre na região monitorada, mantendo no entanto o mesmo padrão médio para consumo de energia total e por operação.

O gráfico da Figura 6.7 mostra a quantidade de bytes enviados pelas redes com ESA localizado (cenário 5) e distribuído (cenário 4). De acordo com os resultados da Tabela 6.7, que ilustra o aumento da quantidade de mensagens, dados e dados relevantes enviados, já era esperado um aumento da quantidade de bytes enviados com o aumento do número de eventos, conforme ilustra o gráfico.

Perda de pacotes: mais uma vez, as métricas relacionadas com a perda de pacotes ilustradas pelas Figuras 6.8(a) e 6.8(b), foram influenciadas pela quantidade de dados e mensagens

Tabela 6.7: Número de mensagens e dados enviados. Resultados obtidos para os cenários 4 (autonômica distribuída) e 5 (autonômica localizada), contendo 200 nós e variando a quantidade de eventos (1, 2 e 4).

	Cenário 4 <i>ESA_{AD4}</i>			Cenário 5 <i>ESA_{AL5}</i>		
Num. Eventos	1	2	4	1	2	4
Msgs	7016.69	8060.82	9113.24	9316.25	9921.85	10788.88
	220.39	247.72	361.31	273.78	398.46	448.86
Dados	11649.91	13663.82	15362.68	14976.41	16066.88	17432.35
	392.97	405.71	602.50	409.89	615.94	687.92
Dados Relev.	2473.88	5486.70	7852.24	3958.91	6293.56	8305.03
	205.53	342.03	557.38	340.56	524.14	655.00

enviadas pelos nós sensores. Com a ocorrência de um número maior de eventos, os nós sensores enviaram maior quantidade de dados e mensagens e conseqüentemente causaram maior sobrecarga no meio de comunicação, causando maior número de colisões e perdas. Os gráficos das figuras mencionadas mostram o aumento linear da porcentagem de perda com o aumento do número de eventos para as redes com ESA distribuído e manutenção de um valor praticamente constante de perda para aquelas com ESA localizado.

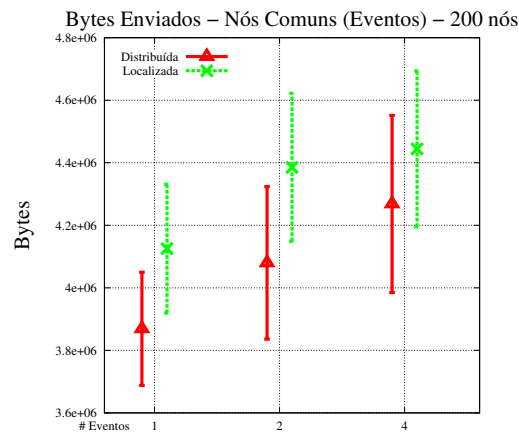
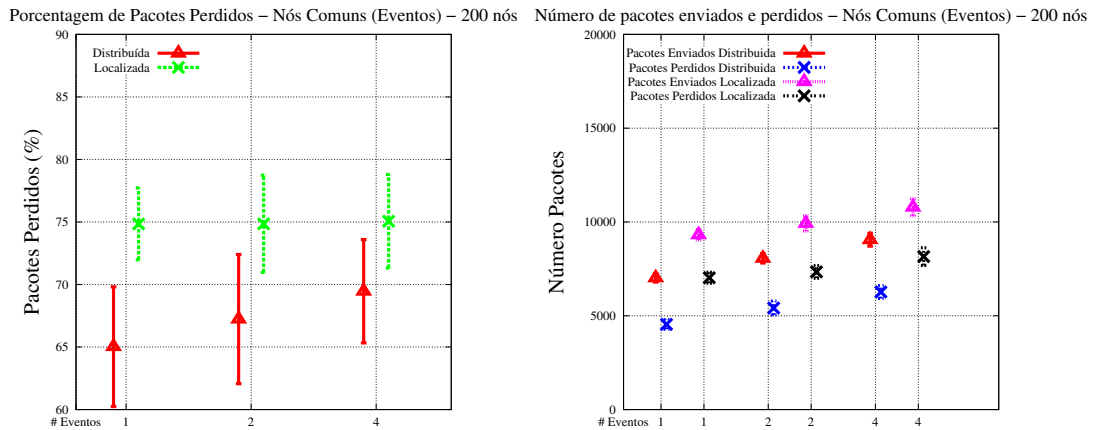


Figura 6.7: Médias e desvios padrão do total de bytes enviados pelos nós comuns. Resultados obtidos para os cenários 4 (autonômica distribuída) e 5 (autonômica localizada), contendo 200 nós sensores e variando a quantidade de eventos na região monitorada (1, 2 e 4).

Dados e Mensagens recebidas pelo PA: para ambos os cenários simulados, os nós sensores foram capazes de entregar uma maior quantidade de dados para o PA quando um maior número de eventos ocorreu na região monitorada, apesar da tendência de elevação da porcen-



(a) Porcentagem de pacotes perdidos.

(b) Número de pacotes enviados e descartados.

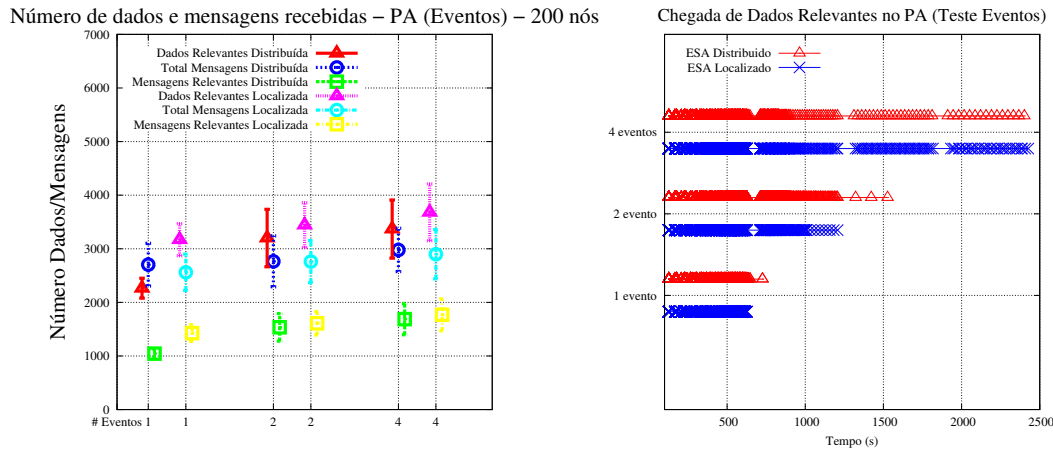
Figura 6.8: Média e desvio padrão relativos às métricas de envio e perda de pacotes para os cenários 4 (autônoma distribuída) e 5 (autônoma localizada).

tagem de pacotes perdidos devido a uma sobrecarga no meio de comunicação. A Figura 6.9(a) mostra os valores obtidos com as simulações para entrega de mensagens, mensagens relevantes e dados relevantes. A Figura 6.9(b) ilustra a chegada de dados relevantes durante a ocorrência dos eventos na região monitorada. O gráfico demonstra a atuação do ESA nos casos em que diferentes números de eventos ocorrem, aumentando as taxa de sensoriamento e disseminação nos momentos adequados e diminuindo as mesmas nos intervalos de ocorrência dos eventos.

Também para esse conjunto de experimentos, a relação entre os resultados obtidos para o cenário 4 (ESAs distribuídos) e cenário 5 (ESAs localizados) se manteve igual aquela apresentada pelas Seções 6.1, 6.2, 6.3 e 6.4.

6.8 Variação da Quantidade de Processamento para Execução do Ciclo Autônomo

Cada iteração do ciclo autônomo corresponde a um determinado conjunto de instruções de máquinas a serem executadas pelo nó sensor. Nos experimentos realizados acima considerou-se que cada iteração correspondia a 40.000 instruções de máquina. Como base de comparação, uma aplicação básica presente no sistema operacional TinyOS para coleta e envio de dados de forma periódica apresenta aproximadamente 30.000 instruções de máquina [TinyOS, 2006]. Uma vez que as instâncias de ESA criadas podem ser de diferentes níveis de complexidade, conforme estabelecido no Capítulo 4, foram realizadas simulações para diferentes valores de quantidade de instruções de máquinas referentes a cada iteração do ciclo autônomo. Foram consideradas iterações com 40.000, 80.000 e 120.000 instruções. Os demais parâmetros das simulações também foram mantidos iguais aqueles descritos pelo Capítulo 5.



(a) Médias e desvios padrão do número de dados e mensagens recebidas pelo PA. (b) Chegada de dados relevantes no PA.

Figura 6.9: Médias e desvios padrão do número de dados e mensagens recebidas pelo PA e produtividade na recepção de dados. Resultados obtidos para os cenários 4 (autônoma distribuída) e 5 (autônoma localizada), contendo 200 nós sensores e variando a quantidade de eventos na região monitorada (1, 2 e 4).

Consumo de Energia: a Tabela 6.8 apresenta o consumo médio de energia total e por operação (linhas brancas) e seus respectivos desvios padrão (linhas em cinza), considerando-se a variação do número de instruções de máquina por iteração do ciclo autônomo. Os valores apresentados para consumo total médio, em ambos os cenários, continuam os mesmos apresentados pela Seção 6.1. Conforme esperado, o consumo de energia médio total apresentado pelos nós apresentou um aumento, ainda que bastante sutil. É notável no entanto o aumento da porcentagem de consumo dos mesmos com processamento, em especial para os nós com ESA localizados, uma vez que os mesmos não são retirados de serviço, ficando ativos durante todo o tempo de simulação. O aumento da porcentagem de consumo com o serviço de processamento é acompanhado por uma diminuição da porcentagem de consumo com comunicação e sensoriamento, para ambos os cenários. Dessa forma, fica demonstrado o impacto do uso de soluções ou algoritmos mais elaborados para a instância de ESA utilizada: apesar de não apresentar impacto muito significativo no consumo total de energia, as atividades de comunicação e sensoriamento ficam prejudicadas, dada o consumo maior da energia disponível pelo serviço de processamento.

Os resultados relativos as outras métricas consideradas nessa dissertação (Seção 5.1) estão apresentadas na Tabela 6.9 a seguir. Mais uma vez, as linhas brancas contém os valores médios obtidos e as linhas em cinza os respectivos desvios padrão. Também nessas simulações, pode-se notar que, para todas as métricas, as mesmas relações entre os cenários 4 (distribuído) e 5 (localizado), apresentadas nas Seções 6.1, 6.2, 6.3 e 6.4 foram mantidas. A quantidade de dados relevantes enviados em todos os casos pode ser considerada bastante similar. A ocorrência de um evento na região monitorada se dá em um momento em que os nós possuem

Tabela 6.8: Consumo médio total de energia dos nós comuns em Joules e porcentagem média do consumo por serviço para os cenários 4 (autônomo distribuído) e 5 (autônomo localizado), contendo 200 nós e variando o número de instruções para ciclo autônomo (40.000, 80.000 e 120.000 instruções).

	Cenário 4 <i>ESA_{AD4}</i>			Cenário 5 <i>ESA_{AL5}</i>		
	40	80	120	40	80	120
Número Instruções (milhares)	40	80	120	40	80	120
Consumo total (J)	0.61	0.63	0.66	0.97	0.99	0.99
	0.03	0.03	0.02	0.01	0.00	0.03
% Comunicação	98.08	96.65	95.21	87.83	81.24	75.52
	0.13	0.13	0.27	0.85	1.24	10.09
% Processamento	1.84	3.30	4.74	11.86	18.46	24.18
	0.09	0.13	0.26	0.84	1.23	10.03
% Sensoriamento	0.05	0.05	0.05	0.31	0.30	0.29
	0.01	0.01	0.01	0.05	0.05	0.08

energia suficiente para aumentarem suas taxas de sensoriamento e disseminação e então enviar maior quantidade de dados para o PA. Nesse ponto, o aumento do consumo dos nós com processamento não apresenta impacto relevante e os nós com 40, 80 ou 120 mil instruções por ciclo podem coletar uma quantidade semelhante de dados. No entanto, o número médio de mensagens e dados enviados diminui com o aumento do número de instruções por iteração do ciclo autônomo. Uma vez que maior quantidade de energia é gasta com o processamento desse ciclo, menor é a quantidade de energia disponível para sensoriamento e comunicação (conforme mostra a Tabela 6.8) e portanto, menor quantidade de mensagens e dados são produzidos ao longo do tempo total de simulação.

A Tabela 6.9 também mostra que o aumento do número de instruções não possui impacto significativo na porcentagem de mensagens perdidas, que permanece basicamente constante em todos os casos. Dessa forma, a quantidade de mensagens, mensagens relevantes e dados relevantes recebidos pelo PA é também semelhante para todas as quantidades de instruções avaliadas.

6.9 Considerações Finais

Neste capítulo foram apresentados os resultados de simulação obtidos nessa dissertação. Conforme descrito na Seção 5.1 do Capítulo 5, as métricas de simulação consideradas foram consumo médio de energia total e por serviço (sensoriamento, processamento e comunicação), precisão de dados enviados e recebidos, perda de mensagens e consumo de memória. A Seção 6.1 mostra que o consumo de energia dos nós sensores que possuem uma instância de ESA implementada é em geral melhor do que aquele apresentado pelos nós não gerenciados. Em particular, quando os ESAs puderam se comunicar e realizar a reserva de recursos da rede, a quantidade média de energia consumida foi ainda menor, mesmo com um grande consumo concentrado no serviço de comunicação, sendo este o tipo de operação que mais consome

Tabela 6.9: Resultados das demais métricas para os cenários 4 (autônomo distribuído) e 5 (autônomo localizado), contendo 200 nós e variando o número de instruções para o ciclo autônomo (40.000, 80.000 e 120.000 instruções).

Número Instruções	Cenário 4 <i>ESA_{AD4}</i>			Cenário 5 <i>ESA_{AL5}</i>		
	40	80	120	40	80	120
Mensagens e Dados Enviados						
Total Msgs	8083.13	7773.83	7417.03	10167.33	9249.66	8396.77
	237.88	204.97	228.21	508.50	348.80	308.85
Total Dados	13404.68	12970.93	12442.61	16437.44	15089.47	13831.81
	406.36	363.86	312.57	742.12	522.92	507.52
Dados Relevantes	5775.92	6136.12	5771.27	7243.41	7200.38	7118.73
	421.01	917.16	377.74	503.19	566.24	685.48
Mensagens e Dados Recebidos						
Total Msgs	2971.37	2959.83	2916.00	2756.82	2719.19	2535.06
	440.62	364.37	379.51	452.35	416.96	354.91
Msgs Relevantes	1907.08	1934.67	1952.70	1721.67	1748.44	1610.50
	296.02	234.10	304.92	328.27	331.65	248.93
Dados Relevantes	3857.38	4006.40	4035.06	3681.74	3713.91	3444.75
	797.57	453.00	548.56	593.14	604.01	477.92
Pacotes Perdidos	64.99	64.54	63.16	75.41	73.50	73.22
	2.53	4.30	4.56	4.93	4.33	3.50

energia dos nós sensores [Hill e Culler, 2002].

A Seção 6.2 mostra outro aspecto positivo observado com o uso do ESA: apesar de enviarem menor quantidade de mensagens e dados totais do que os nós sem ESA, as redes cujos componentes possuem um ESA embutido conseguiram enviar uma quantidade significativamente maior de dados relevantes. Isso se deve ao comportamento adaptativo produzido pela execução do ciclo autônomo, o qual ajusta parâmetros dos nós sensores de acordo com a percepção do ambiente externo e interno. Na Seção 6.3 percebe-se que, apesar dos ESAs distribuídos enviarem uma menor quantidade de mensagens, dados e dados relevantes do que os ESAs localizados, a entrega final para o PA acaba sendo praticamente a mesma. As redes não gerenciadas apresentaram os piores resultados para o recebimento de mensagens. Desse modo, a produtividade das redes com ESA distribuído foi melhor do que das redes com ESA localizado. A perda de pacotes apresentada pela Seção 6.4 explica essa melhor produtividade. Um vez que uma menor quantidade de dados foi enviada ao longo do tempo de vida das RSSFs com ESAs distribuídos, devido a reserva de recursos, um número menor de congestionamento e colisões ocorreu, o que levou a uma menor perda de pacotes e conseqüentemente maior produtividade de entrega.

A Seção 6.5 apresentou a quantidade de memória necessária para se implementar a instância de ESA proposta em nós sensores Mica Motes2. Conforme foi explicado, o ESA foi implementado no sistema operacional TinyOS. Os resultados mostraram que, para uma instância simples, apenas uma pequena quantidade de memória foi necessária, o que permite

constatar a viabilidade de se implementar um elemento autonômico ainda mais elaborado para nós sensores com severas restrições de disponibilidade de memória.

Foram ainda apresentados nesse capítulo outros resultados de simulação, obtidos com a variação do intervalo entre as execuções de iterações do ciclo autonômico, chamado intervalo de monitoração (Seção 6.6), variação da quantidade de eventos ocorridos na região monitorada (Seção 6.7) e variação da quantidade de processamento para cada iteração desse ciclo (Seção 6.8). Nesses casos foram comparados apenas os cenários contendo ESAs embutidos em redes contendo 200 nós sensores (cenários 4 e 5).

As principais conclusões obtidas com esses conjuntos de simulações foram: um intervalo de monitoração pequeno em relação aos intervalos de sensoriamento e disseminação não foi considerado interessante, visto que as constantes modificações realizadas nesses dois últimos podem impedir que os serviços relacionados (coleta e envio) sejam executados. Por outro lado, um intervalo de monitoração grande, apesar de permitir maior número de dados e mensagens enviadas, pode levar a um desgaste maior dos elementos da rede, aumentando consumo de energia e porcentagem de mensagens perdidas. Em relação a variação da quantidade de eventos, os resultados demonstraram a capacidade de adaptação dos ESAs utilizados, sendo que os mesmos foram capazes de controlar parâmetros dos nós sensores de forma a enviar maior quantidade de mensagens, dados e dados relevantes quando o número de eventos aumentou, sem realizar aumento do consumo médio de energia. Finalmente, a variação na quantidade de processamento necessário para se executar uma iteração do ciclo autonômico, demonstrou o cuidado que se deve ter ao se projetar uma instância de ESA. Quanto maior a quantidade de processamento, maior o consumo médio de energia, maior a porcentagem de energia gasta com processamento e assim menor a quantidade de dados e mensagens totais enviadas ao PA.

Capítulo 7

Conclusão

Este trabalho apresentou a aplicação de um novo paradigma para realização de controle e supervisão de sistemas chamado computação autônômica para o ambiente específico de redes. Neste caso, ao implementar todos os serviços, conceitos e características descritos para a computação autônômica, as redes, seus elementos e as ligações entre eles passam a integrar uma rede autônômica. Cada componente destas redes possui embutido em si a menor parte de um sistema autônômico chamada de Elemento Autônômico (EA). O EA executa um laço de controle autônômico, no qual realiza monitoração e análise de parâmetros de seus componentes e do ambiente que o cerca. Em seguida, ele é capaz de planejar ações a serem tomadas no intuito de controlar um ou mais aspectos, tais como falhas ou configuração, e finalmente executa o plano de ações elaborado. Cada elemento autônômico é composto por um gerente autônômico, responsável pela execução do laço de controle e um ou mais elementos gerenciados. No caso das redes autônômicas, cada elemento gerenciado é um dispositivo (computador pessoal, aparelho celular, roteador, PDAs, dentre outros) que compõe a rede em questão. Vale lembrar que individualmente, cada dispositivo (elemento de rede) pode conter sub-sistemas autônômicos, ou seja, ele pode ser composto por hardware ou software controlado por um gerente autônômico. Tais gerentes podem realizar o controle individual de seus elementos gerenciados e em última instância colaboram com o “gerente central” responsável pelo dispositivo como um todo. Em se tratando de redes autônômicas, o interesse está restrito aos processos do elemento autônômico que representa e gerencia cada componente destas redes como um todo. Os EAs de uma rede autônômica apresentam ainda um ciclo de vida particular e um ciclo de relacionamento que permite a comunicação e cooperação entre dois ou mais elementos deste tipo.

Foi apresentada a definição e motivação do uso do paradigma de computação autônômica em um tipo específico de rede sem fio chamada RSSFs e uma proposta de um conjunto de passos a serem seguidos para o desenvolvimento de uma RSSF autônômica: (1) definição ou escolha da arquitetura de gerenciamento, (2) definição do modelo do EA e (3) elaboração dos ciclos de vida e de relacionamento dos EAs da rede. É importante notar como os três passos estão interligados e que a implementação de todos eles leva a construção de uma RSSF autônômica completa. O desenvolvimento de soluções para cada uma das três etapas acima

pode ser considerado um desafio de pesquisa. A principal contribuição desse trabalho foi tratar do desafio relacionado com a segunda etapa da construção de um RSSF autonômica. Um modelo de um EA específico para RSSFs, chamado Elemento Sensor Autonômico (ESA), foi proposto. Para tanto, foram utilizados todos os conceitos definidos pela arquitetura Manna para o gerenciamento específico e integrado de RSSF e o formato genérico de um EA definido em [Kephart e Chess, 2003]. Assim, o conjunto de funções autonômicas específicas para cada serviço do laço autonômico, capaz de embutir um comportamento autonômico nos elementos das RSSFs foi proposto. Por meio dessas funções, o ESA pode controlar e supervisionar nós sensores de forma independente, implementando diversos serviços de auto-gerenciamento, adaptando e aprimorando seu funcionamento através de funções que contém algoritmos para aprendizado de máquina. Também foi proposta uma base de conhecimento para esse modelo, a qual possui dados e informações necessárias ao gerenciamento.

O modelo de ESA foi elaborado considerando-se todas as restrições de recurso que os elementos que compõem uma RSSF apresentam. Dessa forma, somente foram propostas funções simples e indispensáveis para elaboração da solução autonômica. Além disso, a implementação de instâncias do modelo de ESA não requer a implementação de todas as funções, e sim somente daquelas importantes e cabíveis para uma solução em particular. Outro fator importante foi a observação de que o ESA somente armazenará dados e informações e executará algoritmos compatíveis com sua disponibilidade de recursos. Nesse caso, tudo o que em determinado momento for dispensável ou proibitivo, dado o compromisso entre precisão e consumo (energia ou memória, por exemplo), deve ser descartado pelo ESA.

Uma instância do modelo genérico de ESA proposto, específica para a plataforma de nós sensores Mica Motes2, foi elaborada, implementada e avaliada por meio de simulações. Os resultados obtidos mostraram que é interessante embutir serviços e funções autonômicas em uma RSSF, através de um ESA, o qual é capaz de gerenciar os recursos de um nó sensor, utilizando-os de forma estratégica. Especificamente, o ESA foi capaz de diminuir o consumo médio de energia dos nós sensores e o número de mensagens perdidas, além de aumentar a precisão da informação recebida pelo PA nos momentos mais importantes, ou seja, nos intervalos de tempo em que eventos relevantes ocorriam no ambiente monitorado. Também ficou demonstrada a viabilidade de se embutir ESAs na memória de nós sensores. A instância criada como estudo de caso nesse trabalho consumiu apenas aproximadamente 1KB de memória.

A apresentação dos conceitos, modelos, serviços, problemas e objetivos da implementação do paradigma da computação autonômica em redes leva à identificação de várias oportunidades de pesquisa. Neste caso, vários desafios devem ainda ser vencidos e novas estratégias, protocolos, algoritmos, hardware, dentre outros aspectos, propostos para cada tipo de tecnologia de rede e para a integração destas. Todos esses desafios podem ser vistos como trabalhos futuros a serem desenvolvidos para diversos tipos de redes, e em especial para RSSFs. Alguns desses trabalhos futuros são:

- Construção de modelos e abstrações de comportamento;
- Definição de teorias de aprendizado e otimização;

- Definição de teorias de negociação;
- Modelagem estatística automatizada;
- Tratamento de especificações incompletas de domínios e informações;
- Definição, correlação, segurança, validação, distribuição e padronização de políticas;
- Detecção e resolução de conflitos entre políticas;
- Integração de novos elementos;
- Geração automática de limites (*Thresholds*);
- Tratamento da natureza dinâmica e de multi-tarefa dos sistemas autonômicos;
- Desenvolvimento de técnicas de proteção contra ataques;
- Desenvolvimento de *benchmarks* para propriedades de auto-gerenciamento;
- Criação de novas linguagens e metáforas que permitirão que humanos monitorem, visualizem e controlem sistemas autonômicos.

Em relação ao modelo de ESA proposto nesse trabalho, existem ainda desafios a serem trabalhados. Dentre eles podemos citar: avaliação da implementação do modelo em nós sensores reais, aumento do número de parâmetros monitorados, seleção dinâmica de algoritmos a serem executados e itens a serem mantidos na BC e implementação de algoritmos mais elaborados de correlação de eventos e aprendizado de máquina.

A computação autonômica proverá esta nova forma de gerenciamento, permitindo que as próprias máquinas se supervisionem de forma transparente para usuários e administradores, com rapidez, eficiência e baixas taxas de erros. Este paradigma permitirá que as redes sejam sistemas autonômicos (redes autonômicas) robustos, capazes de operar continuamente provendo qualidade de serviço e produtividade dos recursos. No entanto, o desenvolvimento destas redes requer intenso trabalho das comunidades científicas e empresariais, no intuito de se propor, definir, implementar e utilizar de forma ampla, novos modelos, protocolos, serviços e aplicações capazes de atender aos requisitos mínimos e características de cada tipo de rede e ainda prover comunicação eficiente e transparente entre eles. Especificamente, as RSSFs são um tipo especial de redes sem fio que podem se beneficiar do uso do paradigma da computação autonômica para implementação de serviços de gerenciamento.

Apêndice A

Projeto Experimental para Análise Funcional

Uma avaliação de parte da instância do ESA proposta nesse trabalho foi realizada por meio da implementação desta em conjunto com uma aplicação já existente no sistema operacional TinyOS [TinyOS, 2006] chamada *Surge*, utilizado pelos nós sensores Mica Motes 2 [Hill e Culler, 2002]. Apesar de não fornecer indicativos de desempenho, devido à dificuldade de obtenção de parâmetros desse tipo em experimentos reais, essa avaliação permite a observação e análise do funcionamento da solução proposta, sendo assim chamada apenas de análise funcional. Com ela é possível se verificar a viabilidade prática de se implementar uma instância de ESA em elementos reais de uma RSSF.

Para a realização do experimento, foram utilizados 4 nós sensores Mica Mote 2, sendo que um deles permaneceu conectado ao PA ou gateway para permitir a recepção de dados produzidos por outros nós da rede e envio dos mesmos, através de comunicação serial, para um computador pessoal. No momento da realização dos experimentos haviam apenas 4 nós sensores disponíveis para realização do mesmo, o que não prejudica os resultados já que esse número de elementos de rede é suficiente para demonstração do funcionamento da solução de ESA proposta nesse trabalho. Também foram utilizadas 3 placas contendo sensores de temperatura, luminosidade, acelerômetro, magnetômetro e microfone (MTS310 [Crossbow Inc., 2006]). Cada placa foi acoplada a um nó sensor Mica Motes 2 (MPR4x0 [Crossbow Inc., 2006]). Somente o nó conectado ao gateway não recebeu placa de sensores. O sensor utilizado pela aplicação utilizada é o sensor de luminosidade. Além disso, conforme mencionado, foi utilizado um gateway (MIB510 [Crossbow Inc., 2006]) contendo um nó sensor conectado aos pinos de expansão, como forma de ligar a RSSF a um PC, um cabo serial e um cabo de força para fornecer energia ao gateway e ao nó conectado a ele.

Cada nó sensor utilizado no experimento possui embutido em sua memória de programa o sistema operacional TinyOS e uma aplicação específica para coleta e envio de dados de luminosidade chamada *Surge*. A essa aplicação foram adicionados os componentes de software que contém a instância de ESA criada. Também foram utilizados outros dois softwares: um software, chamado *Serial Forwarder*, capaz de receber dados por meio da porta serial de um

PC e repassá-los a uma porta de rede TCP qualquer, e um outro aplicativo Java específico para a aplicação *Surge*, capaz de receber os dados repassados pelo *Serial Forwarder* para a porta TCP e com eles montar uma visualização gráfica da RSSF. Esse último aplicativo exibe o número de mensagens recebidas para cada nó da rede, bem como a taxa de recebimento dessas por segundo. Na aplicação *Surge*, os nós sensores realizam sensoriamento programado e disseminação contínua dos dados de luminosidade coletados. Nos experimentos, inicialmente foram mantidas essas configurações para se testar a aplicação das regras e correlação do ESA sobre o intervalo de sensoriamento. Em seguida, a aplicação foi modificada de forma que os nós passassem a realizar disseminação também programada. Nesse caso, foi possível testar a atuação do ESA sobre o intervalo de comunicação.

O intervalo de monitoração utilizado, isto é, o tempo configurado para a execução de iterações subseqüentes do laço de controle autônomo, foi de 20 segundos. O valor inicial para o intervalo de sensoriamento dos nós era de 2 segundos. O valor base para incremento desse intervalo é de 1 segundo, sendo que o limite máximo é de 10 segundos e o mínimo de 1 segundo. Quando os nós realizam comunicação também programada, o valor inicial do intervalo é de 4 segundos. O valor base de incremento desse intervalo também é de 1 segundo, sendo o máximo valor permitido 20 segundos e mínimo de 2 segundos. O sensoriamento de valores baixos de luminosidade caracteriza a ocorrência de eventos, ou coleta de dados relevantes pelos nós sensores.

O Ciclo Autônomo: para o serviço de monitoração, foram considerados os seguintes parâmetros internos e externos, coletados através de funções de acesso direto ao software do nó sensor e análise de fluxo de dados: número de dados coletados, número de dados relevantes coletados, número de dados enviados pelos vizinhos, número de dados relevantes enviados pelos vizinhos, intervalo de sensoriamento e intervalo de comunicação. Uma vez que a organização da RSSF em uma estrutura de entrega de dados em múltiplos saltos apresentou ser uma tarefa bastante difícil, ao invés de monitorar o fluxo de dados somente com base na estrutura de roteamento, os nós sensores monitoraram todos os fluxo de dados ou mensagens de dados que podiam escutar, ou seja, que eram enviadas dentro dos seus alcances de rádio.

Uma vez monitorados, os parâmetros acima são colocados na BC do nó sensor para serem em seguida recuperados pelo serviço de análise. Esse serviço classifica os valores em altos, médios ou baixos, de acordo com limites pré-estabelecidos e alocados também na BC do nó sensor, realizando em seguida uma correlação simples dos mesmos. Os resultados da correlação também são armazenadas na BC e recuperados no momento oportuno pelo serviço de planejamento. Nesse momento, as políticas consideradas pela instância implementada são avaliadas e a ação mais indicada é passada ao serviço de execução. Por motivos de simplificação, apenas as ações de modificação dos intervalos de sensoriamento e comunicação foram mantidas. De acordo com a ação indicada pelo serviço de planejamento, o serviço de execução aumenta, diminui ou mantém esses intervalos nos nós sensores.

Classificação de dados: uma vez que as regras para correlação utilizadas se baseiam na

quantidade de dados relevantes sensoriados pelo próprio nó e recebidos de seus vizinhos, a análise e classificação desses dois parâmetros é importante para a solução implementada e foi realizada da seguinte forma:

- Se o número de dados relevantes coletados ou recebidos dos vizinhos for maior ou igual a metade do total de dados coletados ou recebidos, então ele é considerado alto. Se esse número é menor do que a metade do total e maior do que zero, ele é considerado médio. Somente se os nós não coletam ou recebem nenhum dado relevante, ou seja o número de dados relevantes coletados ou recebidos é igual a zero, é que o mesmo é considerado baixo.

A estratégia baseada em porcentagens, isto é, a análise baseada no número de dados relevantes sobre o número total coletado ou recebido, utilizada na implementação das simulações, teve que ser abandonada nos experimentos devido a incapacidade dos nós Mica Motes 2 tratarem números com ponto flutuante. A Figura A.1 contém um fluxograma que representa as regras de correlação utilizadas nos experimentos.

Regras para correlação: as seguintes regras foram implementadas para o função de correlação de eventos no serviço de análise:

- SE número de dados relevantes é baixo e número de dados relevantes enviados pelos vizinhos também é baixo e intervalo de sensoriamento menor do que máximo permitido, ENTÃO intervalo de sensoriamento inadequado (baixo);
- SE número de dados relevantes é baixo e número de dados relevantes enviados pelos vizinhos é alto ou médio, ENTÃO intervalo de sensoriamento adequado;
- SE número de dados relevantes é alto ou médio e intervalo de sensoriamento maior do que mínimo permitido, ENTÃO intervalo de sensoriamento inadequado (alto);
- SE número de dados relevantes é baixo e intervalo de disseminação menor do que o máximo permitido, ENTÃO intervalo de comunicação inadequado (baixo);
- SE número de dados relevantes é alto e intervalo de disseminação maior do que o mínimo permitido, ENTÃO intervalo de comunicação inadequado (alto);

Políticas para reprogramação: dados os tipos de resultados para a correlação acima, as seguintes políticas podem ser aplicadas:

- **Condição:** Intervalo de sensoriamento inadequado (baixo);
Ação: Aumentar intervalo até limite de 10 segundos;
- **Condição:** Intervalo de sensoriamento inadequado (alto);
Ação: Diminuir intervalo até limite de 1 segundo;

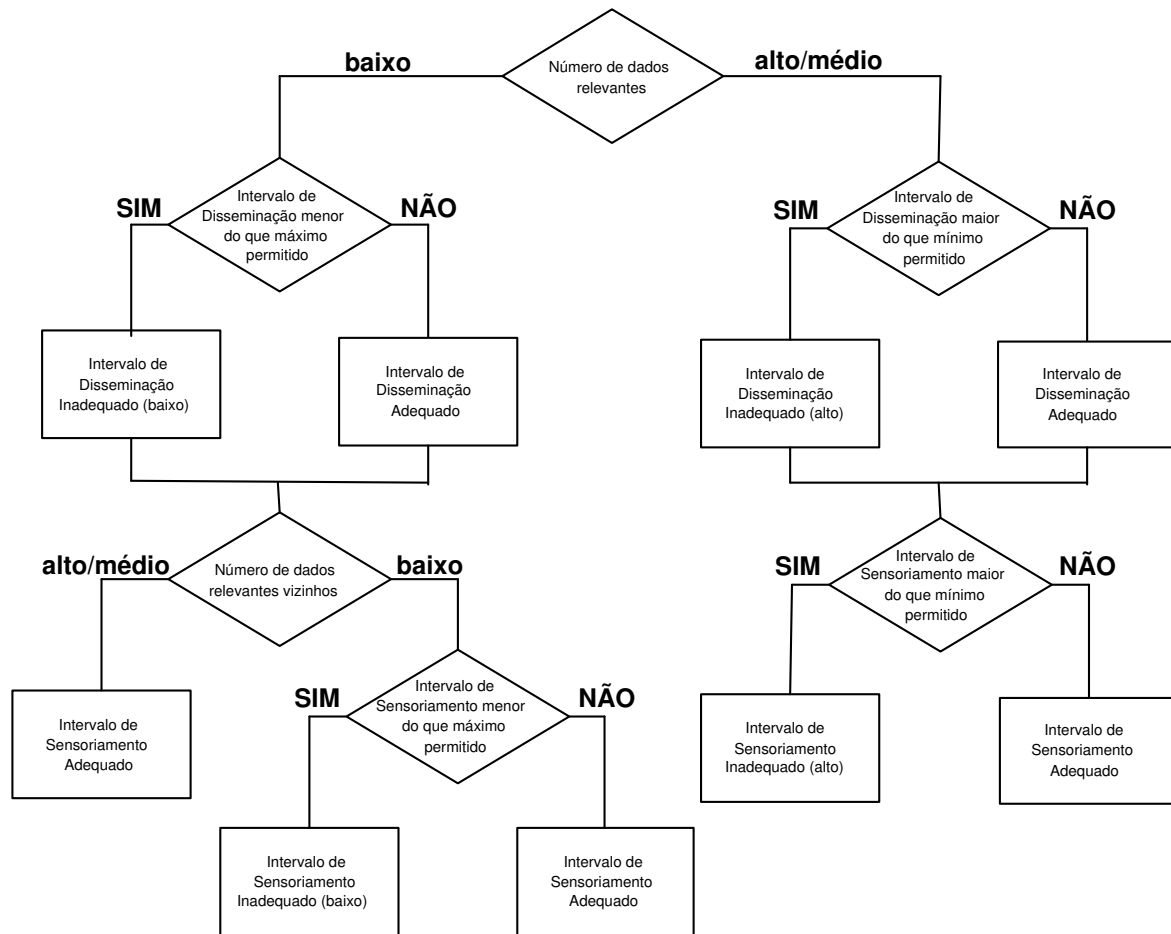


Figura A.1: Fluxograma para regras de correlação utilizadas nos experimentos.

- **Condição:** Intervalo de disseminação inadequado (baixo);
Ação: Aumentar intervalo até limite de 20 segundos;
- **Condição:** Intervalo de disseminação inadequado (alto);
Ação: Diminuir intervalo até limite de 2 segundos;

Dessa forma, analisando variáveis internas e externas, o ESA ajusta os intervalos de sensoriamento e comunicação do nó sensor no qual se encontra embutido. Por meio do auxílio de *Light Emitting Diodes* (LEDS) disponíveis no hardware dos nós Mica Motes 2 e do aplicativo Java desenvolvido para a aplicação *Surge*, pode-se constatar a variação desses intervalos, a cada iteração do ciclo autônomo.

A.1 Cenários Testados e Resultados Obtidos

Análise de Modificação do Intervalo de Sensoriamento: No intuito de avaliar o funcionamento da variação do intervalo de sensoriamento pelo ciclo autônomo descrito acima,

e em especial das regras de correlação e políticas utilizadas, 4 cenários foram propostos e observados durante os experimentos. A seguir estão descritos esses cenários, e apresentadas figuras obtidas com o software Java para visualização da RSSF, que mostram o estado da rede em cada situação. Todas as figuras apresentadas a seguir são referentes a interface gráfica correspondente a aplicação *Surge* utilizada.

Cenário 1: todos os nós sensores contém a placa de sensores, exceto nó do gateway, e são iniciados com intervalo de sensoriamento de 2 segundos e intervalo de monitoração de 20 segundos.

Cenário 2: após alguns instantes sem coletar dados que representem eventos, os nós diminuem suas taxas de sensoriamento, aumentando o intervalo de 1 em 1 segundo, a cada execução do ciclo autonômico, até chegarem no valor máximo permitido de 10 segundos. O nó da base, que não contém placa de sensores, obtem valores relativos a eventos de seu conversor analógico-digital e portanto aumenta sua taxa de sensoriamento, diminuindo o intervalo de 1 em 1 segundo, até chegar ao valor mínimo permitido de 1 segundo.

Cenário 3: a placa de sensores de um dos nós é retirada. Assim como o nó da base, esse elemento da rede começa a receber valores que representam eventos como saída de seu conversor analógico-digital, e começa a aumentar sua taxa de sensoriamento, diminuindo o intervalo que era de 10 segundos (valor máximo) para 1 segundo (valor mínimo).

Cenário 4: o nó sensor que estava sem a placa de sensores no cenário 3, recebe novamente a mesma e começa a coletar dados considerados não relevantes, diminuindo gradativamente sua taxa de sensoriamento. Antes de chegar ao valor mínimo dessa taxa (máximo intervalo de 10 segundos), e ainda coletando dados não relevantes, a placa de sensores de um de seus vizinhos é retirada. Nesse caso, o nó vizinho passa a coletar e enviar valores que representam eventos. O nó sensor então, ao perceber tal situação pára de diminuir sua taxa, mantendo a mesma constante.

Análise de Modificação do Intervalo de Comunicação: Uma vez que apenas duas regras e duas políticas estão relacionadas com a alteração do intervalo de comunicação pelo ESA, apenas 2 cenários foram necessários para verificar o funcionamento destes.

Cenário 1: todos os nós sensores, exceto no conectado ao gateway, são iniciados com a placa de sensores acoplada. Com o decorrer do tempo e as execuções subseqüentes do ESA, o intervalo de comunicação dos nós vai aumentando até atingir o máximo valor permitido de 20 segundos.

Cenário 2: após retiradas as placas de sensores de todos os nós, os mesmos começam a identificar a ocorrência de eventos e diminuem gradativamente o intervalo de sensoriamento até o valor mínimo permitido de 2 segundos.

A.2 Considerações Finais

Esse apêndice apresentou os resultados obtidos com a implementação em plataformas de nós sensores reais de parte da instância de ESA simulada nesse trabalho. Uma vez que é bastante difícil a obtenção de dados de desempenho na realização de experimentos reais, os objetivos dessa implementação eram a avaliação da viabilidade de se implementar o modelo de ESA no hardware da plataforma Mica Motes 2, e verificar que o mesmo segue o comportamento planejado. As figuras apresentadas nesse apêndice ilustram que ambos os objetivos foram atingidos. Por meio do auxílio de LEDS e de uma interface gráfica Java foi possível constatar que o ESA atuou sobre o software dos nós sensores aumentando e diminuindo os intervalos de sensoriamento e disseminação de acordo com a verificação dos ambientes interno e externo, conforme planejado.

Acrônimos

AIB *Application Information Base*

BC Base de Conhecimento

CPB *Communication Protocol Base*

EA Elemento Autônômico

ESA Elemento Sensor Autônômico

ITIL *IT Infrastructure Library*

MIB *Management Information Base*

PA Ponto de Acesso

PBNM *Policy Based Network Management*

QoS *Quality of Service*

RSSF Rede de Sensores Sem Fio

TI Tecnologia da Informação

SNMP *Simple Network Management Protocol*

TMN *Telecommunication Management Network*

CMIP *Common Management Information Protocol*

CMIS *Common Management Information Services*

CMISE *Common Management Information Service Element*

RAM *Random Access Memory*

ROM *Ready-Only Memory*

SLA *Service Level Agreement*

NS *Network Simulator*

LEDS *Light Emitting Diodes*

XML *eXtensible Markup Language*

SO Sistema Operacional

ISO *International Organization for Standardization*

OSI *Open Systems Interconnection*

PDU *Protocol Data Unit*

IETF *Engineering Task Force*

DMTF *Distributed Management Task Force*

PMAC *Policy Management for Autonomic Computing*

KM *Knowledge Management*

Referências Bibliográficas

- [A. Pras and T. Drevers and R. Meent and D. Quartel, 2004] A. Pras and T. Drevers and R. Meent and D. Quartel (2004). Comparing the Performance of SNMP and Web Services-Based Management. *IEEE Transactions on Network and Service Management*, 1(2).
- [AC Blueprint, 2006] AC Blueprint (2006). An Architectural Blueprint for Autonomic Computing. Disponível em <http://www-03.ibm.com/autonomic/pdfs/> - Acessado em Julho de 2006.
- [Agrawal et al., 2005] Agrawal, D.; Lee, K. e Lobo, J. (2005). Policy-based management of networked computing systems. *IEEE Communications Magazine*, 43(10):69–75.
- [AIDE, 2006] AIDE (2006). IBM. Autonomic Integrated Development Environment (AIDE), Disponível em <http://www.alphaworks.ibm.com/tech/aide> - Acessado em Julho de 2006.
- [Akyildiz e Wang, 2005] Akyildiz, I. e Wang, X. (2005). A survey on wireless mesh networks. *IEEE Communications Magazine*, 43(9):23–30. ISSN: 0163-6804.
- [Alavi e Leidner, 1999] Alavi, M. e Leidner, D. E. (1999). Knowledge management systems: issues, challenges, and benefits. *Communications of the AIS*, 1(2):1.
- [Alphaworks, 2006] Alphaworks (2006). IBM. Alphaworks Autonomic Computing, Disponível em <http://www.alphaworks.ibm.com/autonomic> - Acessado em Julho de 2006.
- [Appavoo et al., 2003] Appavoo, J.; Hui, K.; Soules, C. A. N.; Wisniewski, R. W.; Silva, D. M. D.; Krieger, O.; Auslander, M. A.; Edelsohn, D. J.; Gamsa, B.; Ganger, G. R.; McKenney, P.; Ostrowski, M.; Rosenburg, B.; Stumm, M. e Xenidis, J. (2003). Enabling autonomic behavior in systems software with hot swapping. *IBM Systems Journal*, 42(1):60–76.
- [Assunção e Ruiz, 2006] Assunção, H. P. e Ruiz, L. B. (2006). A service management approach for self-healing wireless sensor networks. In *Proceedings of the 1st Autonomic Networking (AN'06)*, Paris, França. Lecture Notes in Computer Science. Springer-Verlag. (Para Aparecer).
- [Autonomic Toolkit, 2006] Autonomic Toolkit (2006). IBM. Autonomic Computing Toolkit (ACT), Disponível em <http://www-128.ibm.com/developerworks/autonomic/probdet.html> - Acessado em Julho de 2006.

- [Babcock et al., 2002] Babcock, B.; Babu, S.; Datar, M.; Motwani, R. e Widom, J. (2002). Models and issues in data stream systems. In *Proceedings of 21st ACM Symposium on Principles of Database Systems (PODS'02)*, pp. 1–16, Madison, Wisconsin, USA. ISBN:1-58113-507-6.
- [Barlow e Edwards, 2001] Barlow, G. J. e Edwards, M. A. (2001). Self-evolving hardware. Technical Report NCSU-CRIM-TR-2001-01, North Carolina State University, Raleigh, NC.
- [Bluetooth, 2006] Bluetooth (2006). IEEE 802.15.1: bluetooth. Disponível em <http://www.bluetooth.com> - Acessado em Julho de 2006.
- [Braga et al., 2004] Braga, T. R. M.; Silva, F.; Ruiz, L. B. e Nogueira, J. M. S. (2004). Manna-Sim: a framework to the simulation of wireless sensors networks (in portuguese). *Electronics Magazine of Undergraduate Scientific Research of the Brazilian Computer Science Society (REIC)*.
- [Braga et al., 2006a] Braga, T. R. M.; Silva, F. A.; Ruiz, L. B. e A., H. P. (2006a). Redes autônomicas. In *Proceedings of 24th Simpósio Brasileiro de Redes de Computadores (SBRC'06)*, Curitiba, Paraná, Brasil. Minicurso.
- [Braga et al., 2006b] Braga, T. R. M.; Silva, F. A.; Ruiz, L. B.; Nogueira, J. M. S. e Loureiro, A. A. F. (2006b). Design and evaluation of an autonomic sensor element. In *Proceedings of 1st Latin American Autonomic Computing Symposium (LAACS'06)*, Campo Grande, Mato Grosso do Sul, Brasil.
- [Braga et al., 2006c] Braga, T. R. M.; Silva, F. A.; Ruiz, L. B.; Nogueira, J. M. S. e Loureiro, A. A. F. (2006c). Um elemento autônomico para redes de sensores sem fio. In *Proceedings of 24th Simpósio Brasileiro de Redes de Computadores (SBRC'06)*, Curitiba, Paraná, Brasil.
- [Bulusu et al., 2001] Bulusu, N.; Estrin, D.; Girod, L. e Heidemann, J. (2001). Scalable coordination for wireless sensor networks: Self-configuring localization systems. In *Proceedings of the 6th IEEE International Symposium on Communication Theory and Application*, p. to appear, St. Martin's College, Ambleside, Cumbria, UK. USC/Information Sciences Institute, IEEE.
- [Cerpa e Estrin, 2004] Cerpa, A. e Estrin, D. (2004). Ascent: Adaptive self-configuring sensor networks topologies. *IEEE Transactions on Mobile Computing*, 3(3):272–285.
- [Chadha et al., 2002] Chadha, R.; Lapiotis, G. e Wright, S. (2002). Policy-based networking. *IEEE Network*, 16(2):8–9. (Guest Editorial).
- [Chen et al., 1996] Chen, M.-S.; Han, J. e Yu, P. S. (1996). Data mining: An overview from a database perspective. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):866–883. ISSN: 1041-4347.
- [Chess et al., 2003] Chess, D. M.; Palmer, C. C. e White, S. R. (2003). Security in an autonomic computing environment. *IBM Systems Journal*, 42(1):107–118.

- [Crossbow Inc., 2006] Crossbow Inc. (2006). Crossbow Technology Inc. - Disponível em <http://www.xbow.com> - Acessado em Julho de 2006.
- [Damianou et al., 2002] Damianou, N.; Bandara, A. K.; Sloman, M. e Lupu, E. (2002). A survey of policy specification approaches. <http://www.doc.ic.ac.uk/~mss/MSSPubs.html> - Acessado em Julho de 2006. Department of Computing, Imperial College of Science Technology and Medicine.
- [(DMTF), 1999] (DMTF), D. M. T. F. (1999). Common Information Model (CIM) Specification. Disponível em <http://www.dmtf.org/standards/cim/> - Acessado em Julho de 2006.
- [DSI, 2006] DSI (2006). Microsoft. Dynamic Systems Initiative (DSI), Disponível em <http://www.microsoft.com/windowsserversystem/dsi/> - Acessado em Julho de 2006.
- [Elnahrawy, 2003] Elnahrawy, E. (2003). Research directions in sensor data streams: Solutions and challenges. Technical Report DCIS-TR-527, Rutgers University, New Jersey, USA.
- [Estrin et al., 2002] Estrin, D.; Sayeed, A. e Srivastava, M. (2002). Wireless sensor networks. In *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*, Atlanta, USA. Tutorial.
- [Eymann et al., 2003] Eymann, T.; Reinicke, M.; Ardaiz, O.; Artigas, P.; Freitag, F. e Navarro, L. (2003). Self-organizing resource allocation for autonomic networks. In *Proceedings of the 14th International Workshop on Database and Expert Systems Applications (DEXA'03)*, p. 656, Prague, Czech Republic. Lectures Notes in Computer Science - ISBN:3-540-40806-1.
- [Ferrière et al., 2006] Ferrière, H.; Meier, R.; Fabre, L. e Metrailler, P. (2006). TinyNode: A Comprehensive Platform for Wireless Sensor Network Applications. In *Information Processing in Sensor Networks (IPSN 2006)*, Nashville, USA.
- [Figueiredo et al., 2004] Figueiredo, C. M.; Nakamura, E. F. e Loureiro, A. A. (2004). Multi: A hybrid adaptive dissemination protocol for wireless sensor networks. In *Proceedings of the 1st International Workshop on Algorithmic Aspects of Wireless Sensor Networks (Algosensors'04)*, volume 3121, pp. 171–186, Turku, Finland. Springer.
- [Figueiredo et al., 2005] Figueiredo, C. M. S.; Santos, A. L.; Loureiro, A. A. e Nogueira, J. M. S. (2005). Policy-based adaptive routing in autonomous wsns. In *Proceedings of the 16th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM'05)*, Barcelona, Spain.
- [Fioreze et al., 2005] Fioreze, T.; Granville, L. Z.; Almeida, M. J. B. e Tarouco., L. M. R. (2005). Comparing web services with snmp in a management by delegation environment. In *IFIP/IEEE International Symposium on Integrated Network Management (IM 2005)*, pp. 601–614, Nice, França. ISBN: 0-7803-9087-1.

- [Ganek e Corbi, 2003] Ganek, A. G. e Corbi, T. A. (2003). The dawning of the autonomic computing era. *IBM Systems Journal*, 42(1):5–18.
- [Hauser, 2000] Hauser, J. R. (2000). *Augmenting a Processor with Reconfigurable Hardware*. PhD thesis, University of California, Berkeley, California, USA.
- [Hill e Culler, 2002] Hill, J. e Culler, D. (2002). Mica: A wireless platform for deeply embedded networks. *IEEE Micro*, 22(6):12–24. ISSN: 0272-1732.
- [Hollar, 2000] Hollar, S. E.-A. (2000). *Cots dust*. Master's thesis, Department of Engineering-Mechanical Engineering - University of California, Berkeley.
- [ISO, 2006] ISO (2006). International Organization for Standardization (ISO). Disponível em <http://www.iso.org/iso/en/ISOOnline.frontpage> - Acessado em Julho de 2006.
- [ITIL, 2006] ITIL (2006). IT Infrastructure Library. Disponível em <http://www.itil.co.uk/> - Acessado em julho de 2006.
- [Ju et al., 2000] Ju, H.-T.; Choi, M.-J. e Hong, J. W. (2000). An efficient and lightweight embedded web server for web-based network element management. *Int. J. Netw. Manag.*, 10(5):261–275. ISSN: 1099-1190.
- [Kephart, 2005] Kephart, J. O. (2005). Research challenges of autonomic computing. In *27th International Conference on Software Engineering (ICSE'05)*, pp. 15–22, St. Louis, MO, USA.
- [Kephart e Chess, 2003] Kephart, J. O. e Chess, D. M. (2003). The vision of autonomic computing. *IEEE Computer*, 36(1):41–50.
- [Krishnan, 2004] Krishnan, R. (2004). *Efficient self-organization of large wireless sensor networks*. PhD thesis, Boston University, College of Engineering. Major Professor-David Starobinski.
- [Lanfranchi et al., 2003] Lanfranchi, G.; Peruta, P. D.; Perrone, A. e Calvanese, D. (2003). Toward a new landscape of systems management in an autonomic computing environment. *IBM Systems Journal*, 42(1):119–128.
- [L.C. e A.A., 2001] L.C., I. e A.A., C. (2001). Projeto de pesquisa - parte VIII: método estatístico / análise estatística. Disponível em <http://www.evidencias.com/planejamento/pdf/> - Acessado em Julho de 2006.
- [Levis et al., 2003] Levis, P.; Lee, N.; Welsh, M. e Culler, D. (2003). Tossim: accurate and scalable simulation of entire tinyos applications. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys'03)*, pp. 126–137, Los Angeles, CA, USA.

- [Lim, 2006] Lim, J.-J. (2006). System support for management of networked low-power sensors. In *Proceedings of the 10th IEEE/IFIP Network Operation and Management Symposium (NOMS'06)*, Vancouver, Canada.
- [Littman et al., 2004] Littman, M. L.; Ravi, N.; Fenson, E. e Howard, R. (2004). Reinforcement learning for autonomic network repair. In *Proceedings of the 1st International Conference on Autonomic Computing (ICAC'04)*, pp. 284–285, New York, USA.
- [Liu e Martonosi, 2003] Liu, T. e Martonosi, M. (2003). Impala: A middleware system for managing autonomic parallel sensor systems. In *Proceedings of the 9th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP'03)*, pp. 107–118, San Diego, CA, USA.
- [Lopes et al., 2006] Lopes, C.; Melo, J.; Assunção, H.; Braga, T.; Silva, F.; Ruiz, L.; Loureiro, A. e Nogueira, J. (2006). Mannasim: Simulando redes de sensores sem fio. In *XXIV Simpósio Brasileiro de Redes de Computadores (SBRC'06)*, Curitiba, PR, Brasil. Salão de Ferramentas.
- [Lymberopoulos e Savvides, 2005] Lymberopoulos, D. e Savvides, A. (2005). XYZ: A motion-enabled, power aware sensor node platform for distributed sensor network applications. In *Proceedings of Information Processing in Sensor Networks (IPSN 2005)*, Los Angeles, CA, USA.
- [M. et al., 2000] M., R.; F., T. e C., A. (2000). Dynamic voltage scaling techniques for distributed microsensor networks. In *Proceedings of the IEEE Computer Society Annual Workshop on VLSI WVLSI'00*, Orlando, FL, USA.
- [Machado et al., 2005] Machado, M.; Goussevskaia, O.; Mini, R.; Rezende, C.; Loureiro, A.; Mateus, G. e Nogueira, J. (2005). Data dissemination in autonomic wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, 23(12):2305–2319.
- [Marsh et al., 2004] Marsh, D.; Tynan, R.; O’Kane, D. e O’Hare, G. M. P. (2004). Autonomic wireless sensor networks. *Engineering Applications of Artificial Intelligence*, pp. 741–748. Elsevier Publishers.
- [McElroy e W, 2002] McElroy e W, M. (2002). *The New Knowledge Management : Complexity, Learning, and Sustainable Innovation*. Butterworth-Heinemann. ISBN: 0750676086.
- [Meira, 1997] Meira, D. M. (1997). *A Model for Alarm Correlation in a Telecommunication Network*. PhD thesis, Computer Science Department of the Federal University of Minas Gerais, Belo Horizonte, MG, Brasil.
- [Melcher e Mitchell, 2004] Melcher, B. e Mitchell, B. (2004). Towards an autonomic framework: Self-configuring network services and developing autonomic applications. *Intel Technology Journal*, 8(4):279–290. ISSN: 1535-864X.

- [Mitchell, 1997] Mitchell, T. (1997). *Machine Learning*. McGraw Hill. ISBN: 0070428077.
- [MoteIV Corp., 2006] MoteIV Corp. (2006). Disponível em <http://www.moteiv.com/> - Acessado em Julho de 2006.
- [N1 Plan, 2006] N1 Plan (2006). Sun microsystems. N1 plan, Disponível em <http://www.sun.com/software/n1gridsystem/index.xml> - Acessado em Julho de 2006.
- [Network Simulator, 2006] Network Simulator (2006). UCB/LBNL/VINT network simulator (ns-2). Disponível em <http://www.isi.edu/nsnam/ns> - Acessado em Julho de 2006.
- [Node, 2006] Node, B. (2006). Body Sensor Network Node - Disponível em <http://www.doc.ic.ac.uk/vip/ubimon/index.html> - Acessado em Julho de 2006.
- [Norman et al., 2003] Norman, D. A.; Ortony, A. e Russel, D. M. (2003). Affect and machine design: Lessons for the development of autonomous machines. *IBM Systems Journal*, 42(1):38–44.
- [P. e K., 2000] P., G. J. e K., W. J. (2000). Wireless integrated network sensors. *Communications of the ACM*, 43(5):51–58. ISSN 0001-0782.
- [Perkins e McGinnis, 1996] Perkins, D. T. e McGinnis, E. (1996). *Understanding SNMP MIB*. Prentice Hall.
- [Pujolle e Chaouchi, 2005] Pujolle, G. e Chaouchi, H. (2005). An autonomic oriented architecture for wireless sensor networks. *Annals of Telecommunications - Sensor Networks*, 60(6/7):819–830.
- [Ruiz, 2003] Ruiz, L. B. (2003). *Manna: A Management Architecture for Wireless Sensor Networks*. PhD thesis, Computer Science Department of the Federal University of Minas Gerais, Belo Horizonte, MG, Brasil.
- [Ruiz et al., 2004] Ruiz, L. B.; Nogueira, J. M. S. e Loureiro, A. A. F. (2004). Sensor network management. In Ilyas, M. e Mahgoub, I., editores, *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, volume 1, chapter 3. CRCPress LLC., Flórida, FL, USA. ISBN 0-8493-1968-4.
- [Russel et al., 2003] Russel, L. W.; Morgan, S. P. e Chron, E. G. (2003). Clockwork: A new movement in autonomic systems. *IBM Systems Journal*, 42(1):77–84.
- [ScatterWeb, 2006] ScatterWeb (2006). Disponível em <http://www.scatterweb.net/> - Acessado em Julho de 2006.
- [Secretaria Municipal de Governo, 2006] Secretaria Municipal de Governo (2006). Lei número 8.262 de 04 de dezembro de 2001. Monitoramento e Controle do Ar no Município de Belo Horizonte, Disponível em www.pbh.gov.br/smsa/biblioteca/gevis/lei_8262.pdf - Acessado em Julho de 2006.

- [SensorNet, 2006] SensorNet (2006). Grupo de Pesquisa em RSSFs - DCC/UFMG. Disponível em <http://www.sensornet.dcc.ufmg.br> - Acessado em Julho de 2006.
- [Silva et al., 2004a] Silva, F.; Braga, T. R.; Ruiz, L. B. e Nogueira, J. M. S. (2004a). Wireless sensor nodes technology (in portuguese). *Control & Instrumentation Magazine*, 10(92):76–87.
- [Silva et al., 2004b] Silva, F.; Ruiz, L. B.; Braga, T. R.; Nogueira, J. M. S. e Loureiro, A. A. F. (2004b). Designing a self-organizing wireless sensor network. In *Proceedings of the 1st International Workshop on Mobility Aware Technologies and Applications (MATA'04)*, pp. 186–195, Florianópolis, SC, Brasil. Springer-Verlag, v.3284.
- [Silva et al., 2005] Silva, F.; Ruiz, L. B.; Braga, T. R. M.; Nogueira, J. M. S. e Loureiro, A. A. F. (2005). Defining a wireless sensor network management protocol. In *Proceedings of the 4rd Latin American Network Operation and Management Symposium (LANOMS'05)*, Porto Alegre, RS, Brasil.
- [Sloman, 1994] Sloman, M. (1994). *Network and Distributed Systems Management*. Addison-Wesley.
- [Sohrabi et al., 2000] Sohrabi, K.; Gao, J.; Ailawadhi, V.; e Pottie, G. (2000). Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications*, 7(5):16–27.
- [Sowa, 2000] Sowa, J. F. (2000). *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. MIT Press. ISBN: 0-534-94965-7.
- [Stallings, 1998] Stallings, W. (1998). *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*. Addison-Wesley.
- [Stojanovic et al., 2004] Stojanovic, L.; Schneider, J.; Maedche, A.; Libischer, S.; Studer, R.; Lumpp, T.; Abecker, A.; Breiter, G. e Dinger, J. (2004). The role of ontologies in autonomic computing systems. *IBM Systems Journal*, 43(3):598–616.
- [Tian et al., 2005] Tian, W.; Zulkernine, F.; Zebedee, J.; Powley, W. e Martin, P. (2005). Architecture for an autonomic web service environment. In *Proceedings of the Joint Workshop on Web Services and Model-Driven Enterprise Information System (WSMDEIS'05)*, pp. 54–66, Miami,FL,USA.
- [TinyOS, 2006] TinyOS (2006). TinyOS operating system. Disponível em <http://www.tinyos.net> - Acessado em Julho de 2006.
- [TMN, 1992] TMN (1992). CCITT: Recommendation M.3010, Principles for a Telecommunications Management Network.
- [UDC, 2006] UDC (2006). Hewlett-packard (HP). Utility Data Center (UDC), Disponível em <http://www.managementsoftware.hp.com/service/index.html> - Acessado em Julho de 2006.

- [VALUMO, 2006] VALUMO (2006). NEC. Valumo), Disponível em <http://www.sw.nec.co.jp/valumo/english/platform/index.html> - Acessado em Julho de 2006.
- [Velayos e Karlsson, 2005] Velayos, H. e Karlsson, G. (2005). Autonomic networking for wireless LANs. In *IEEE Infocom 2005 Student Workshop*, Miami, FL, USA.
- [Wang et al., 2006] Wang, D.; Zhang, Q. e Liu., J. (2006). Self-protection for wireless sensor networks. In *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems (ICDCS'06)*, Lisboa, Portugal.
- [Warneke et al., 2001] Warneke, B.; Last, M.; Liebowitz, B. e Pister, K. (2001). Smart dust: communicating with a cubic-millimeter computer. *Computer*, 34(1):44–51. ISSN: 0018-9162.
- [Weatherall e Jones, 2002] Weatherall, J. e Jones, A. (2002). Ubiquitous networks and their applications. *IEEE Wireless Communications*, 9(1):18–29. ISSN: 1536-1284.
- [Weiss e Kulikowski, 1991] Weiss, S. e Kulikowski, C. (1991). *Computer Systems That Learn*. Morgan Kaufmann. ISBN: 1-55860-065-5.
- [White et al., 2005] White, T.; Calvert, D. e Litkey, J. (2005). Design of an autonomic element for server management. In *Proceedings of the 2nd IEEE International Conference on Autonomic Computing (ICAC'05)*, Seattle, WA, USA.
- [Wi-fi, 2006] Wi-fi (2006). IEEE 802.11: wireless fidelity (Wi-fi). Disponível em <http://www.wi-fi.org> - Acessado em Julho de 2006.
- [WiMax, 2006] WiMax (2006). IEEE 802.16: worldwide interoperability for microwave access (WiMax). Disponível em <http://www.wimaxforum.org> - Acessado em Julho de 2006.
- [Zhang e Arora, 2002] Zhang, H. e Arora, A. (2002). Gs3: scalable self-configuration and self-healing in wireless networks. In *PODC'02: Proceedings of the twenty-first annual symposium on Principles of distributed computing*, pp. 58–67, New York, NY, USA. ACM Press. ISBN: 1-58113-485-1.
- [ZigBee, 2006] ZigBee (2006). IEEE 802.15.4 - zigbee alliance. Disponível em <http://www.zigbee.com> - Acessado em Julho de 2006.
- [Zimmermann et al., 2005] Zimmermann, K.; Felis, S.; Schmid, S.; Eggert, L. e Brunner, M. (2005). Autonomic wireless network management. In *Workshop on Autonomic Communication (WAC'05)*, pp. 57–70, Atenas, Grécia.