

UNIVERSIDADE FEDERAL DO CEARÁ
DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA
PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA

Máquina de Inferência Autônoma Distribuída para RSSF

Nídia Glória da Silva Campos

FORTALEZA – CEARÁ
AGOSTO 2010



UNIVERSIDADE FEDERAL DO CEARÁ
DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA
PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA

Máquina de Inferência Autônoma Distribuída para RSSF

Autor

Nídia Glória da Silva Campos

Orientador

Prof. Dr. José Neuman de Souza

Co-Orientador

Prof. Dr. Danielo Gonçalves Gomes

*Dissertação de Mestrado apresentada
à Coordenação do Curso de
Pós-Graduação em Engenharia de
Teleinformática da Universidade
Federal do Ceará como parte dos
requisitos para obtenção do grau
de Mestre em Engenharia de
Teleinformática.*

FORTALEZA – CEARÁ

AGOSTO DE 2010

C215m Campos, Nídia Glória da Silva
Máquina de inferência autônômica distribuída para RSSF / Nídia Glória da Silva Campos. -- Fortaleza, 2010.
76 f. ; il. enc.

Orientador: Prof. Dr. José Neuman de Souza
Co-orientador: Prof. Dr. Danielo Gonçalves Gomes
Área de concentração: Sinais e Sistemas (SS)
Dissertação (Mestrado) - Universidade Federal do Ceará, Centro de Tecnologia, Depto. de Engenharia de Teleinformática, Fortaleza, 2010.

1. RSSF. 2. Autoconfiguração. 3. Lógica Fuzzy. I. Souza, José Neuman (Orient.). II. Gomes, Danielo Gonçalves (Co-orient.). III. Universidade Federal do Ceará – Programa de Pós-Graduação em Engenharia de Teleinformática. IV. Título.

CDD 621.38

NÍDIA GLÓRIA DA SILVA CAMPOS

Máquina de Inferência Autônômica Distribuída para RSSF

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia de Teleinformática e aprovada em sua forma final pelo programa de Pós Graduação em Engenharia de Teleinformática da Universidade Federal do Ceará.
Banca Examinadora:

Prof. Dr. José Neuman de Souza
Orientador

Prof. Dr. Danielo Gonçalves Gomes
Co-Orientador

Prof^a. Dr^a. Rossana M. de Castro Andrade
Universidade Federal do Ceará

Prof. Dr. Ângelo Roncalli Alencar Brayner
Universidade de Fortaleza

Fortaleza, 12 de agosto de 2010

Resumo

Redes de Sensores Sem Fio (RSSF) são exemplos de *Resource-Constrained Networks* (RCNs) nas quais recursos de processamento, armazenamento e energia são limitados. Esse tipo de rede pode fornecer informações de contexto a sistemas de inteligência de ambiente, os quais tendem à sobrecarga conforme aumenta-se a quantidade de nós sensores da RSSF e/ou o grau de heterogeneidade dos seus tipos de dados capturados. Este trabalho propõe uma máquina de inferência automática distribuída (MIAD) que usa lógica *fuzzy* para aumentar o nível semântico de informações de contexto da RSSF, além de autoconfigurar intervalos de sensoriamento, disseminação dos nós sensores e a redundância de mensagens de contexto da área monitorada. Testes experimentais realizados com sensores de temperatura e umidade relativa mostram que a MIAD embarcada nos nós sensores propicia aumento de 21,8% no envio de informações de contexto relevantes sobre risco de fogo, bem como diminui o consumo de energia da RSSF em 18,4%. A MIAD apresentou melhores resultados quando comparada tanto a uma aplicação distribuída produtora de contexto para RSSF quanto a um motor autônomo baseado em regras rígidas (*crisp rules*) responsável pela autoconfiguração da RSSF.

Palavras-chave: RSSF, contexto, autoconfiguração, lógica *fuzzy*.

Abstract

Wireless sensor networks are examples of resource constrained networks (RCNs) in which processing resources, storage and energy are limited. This type of network can provide context information to Ambient Intelligence systems, which tend to overload as it increases the amount of sensor nodes in WSN and/or the degree of heterogeneity of its data types captured. This paper proposes an autonomic distributed inference machine (MIAD) that uses fuzzy logic to increase the level of semantic information in the context of WSN, and self-configures sensing intervals, dissemination of sensor nodes and redundancy message context of the monitored area. Experimental tests carried out with temperature sensors and relative humidity show that MIAD embedded in sensor nodes provides an increase of 21.8% in dispatching context information relevant to fire risk and reduces the power consumption of WSN in 18.4%. MIAD has shown better results when compared to both the producer of a distributed application framework for WSN as well as an autonomic motor based on crisp rules responsible for self-configuration of WSN.

Key Words: Autonomic Computing; WSN; Fuzzy Logic; Self-Configuration.

Aos meus pais.

Agradecimentos

Agradeço a Deus por toda paciência, ousadia e perseverança dadas e que me ajudaram a iniciar e concluir este trabalho.

À toda minha família, especialmente meus pais, Neide e Antônio, pelo amor, carinho, dedicação e incentivos durante toda minha vida.

Aos meus orientadores, professores Neuman e Danielo, pela confiança, motivação, apoio e paciência nesses anos de formação acadêmica.

Aos professores Rossana e Ângelo, pela presença na banca examinadora e valiosas sugestões e críticas.

A todos funcionários do GREat, pela atenção e suporte às minhas atividades. Aos colegas do GREat, em especial, Atslands, Sandra e Carlos Giovanni pelos diversos comentários e, principalmente, pela amizade.

À CAPES, pelo apoio financeiro no meu primeiro ano de estudos e aos chefes e colegas do Instituto Federal de Educação, Ciência e Tecnologia do Ceará - IFCE, pela compreensão e ajuda oportuna no restante do mestrado.

Ao CNPq, pelo apoio financeiro para apresentação de artigo no *International Telecommunications Symposium* - ITS 2010.

À Comunidade Católica Salve Rainha, por me ensinar que a fé e a razão podem caminhar juntas. Aos meus irmãos de Caminhada, que assumiram a minha missão durante minhas ausências.

*"A paciência tudo alcança."
Santa Tereza D'Ávila*

Sumário

Lista de Figuras	viii
Lista de Tabelas	ix
Lista de Siglas	x
1 Introdução	1
1.1 Contextualização	1
1.2 Motivação	2
1.3 Objetivo	3
1.3.1 Objetivo geral	3
1.3.2 Objetivos específicos	3
1.4 Metodologia	4
1.5 Organização do documento	5
2 Revisão Bibliográfica	6
2.1 RSSF e seus principais componentes	6
2.2 RSSF e a Inteligência de Ambiente	9
2.3 RSSF e Sistemas Sensíveis ao Contexto	11
2.4 Inteligência Artificial aplicada a RSSF	15
2.5 RSSF e a Computação Autônoma	18
2.5.1 Computação Autônoma	19
2.5.2 RSSF Autônomas	19
2.6 RSSF e Sistemas de Controle <i>Fuzzy</i>	23
2.7 Conclusão	24
3 Proposta: MIAD	26
3.1 Descrição Geral da MIAD	26
3.2 Gerente Autônomo - MIA	27
3.2.1 Serviço de Monitoração	28
3.2.2 Serviço de Análise	30
3.2.3 Serviço de Planejamento	31
3.2.4 Serviço de Execução	33

3.3	Elemento Gerenciado	33
3.4	Conclusão	35
4	Resultados	37
4.1	Descrição das Máquinas de Inferência	37
4.2	Descrição do Cenário de Testes	38
4.3	Análise dos resultados	40
4.4	Conclusão	44
5	Considerações Finais	45
5.1	Conclusões	45
5.2	Contribuições	46
5.3	Trabalhos Futuros	48
	Apêndice A Base de Conhecimento de Autoconfiguração	50
	Apêndice B Base de Conhecimento de Risco de Fogo	53
	Referências Bibliográficas	64

Lista de Figuras

2.1	Componentes de uma RSSF	8
2.2	Cenário do e-SENSE	9
2.3	Níveis de Abstração do SENSEI	10
2.4	Arquitetura Conceitual em Camadas	11
2.5	Estrutura de um Elemento Autônomo	20
2.6	Estrutura de um Elemento Sensor Autônomo	22
3.1	Máquina de Inferência Autônoma Distribuída - MIAD	27
3.2	Conj. <i>fuzzy</i> da perc. dados relevantes e da perc. energia residual . . .	29
3.3	Conj. <i>fuzzy</i> do intervalo de sensoriamento e de disseminação	30
3.4	Conj. <i>fuzzy</i> da perc. vizi. config. semelhante e da redundancia	30
3.5	Conj. <i>fuzzy</i> autoconfig. do inter. sensoriamento e de disseminação . .	31
3.6	Conj. <i>fuzzy</i> de temperatura e da umidade relativa	34
3.7	Conj. <i>fuzzy</i> do quantificador <i>most</i> e do risco de fogo	34
4.1	Testbed	39
4.2	Média dos intervalos de sensoriamento e disseminação	41
4.3	Mensagens de dados enviadas por cada nó sensor com ESA	42
4.4	Mensagens de contexto enviadas por cada nó sensor com a MIAD . .	43
4.5	Méd. do coef. de variância dos inter. de disseminação e sensoriamento	43
4.6	Média de bytes das mensagens enviadas ao sistema de monitoração .	44

Lista de Tabelas

3.1	Exemplo de um plano de ações	33
4.1	Tipos de Mensagens	38
4.2	Média dos Parâmetros Analisados	41
A.1	Regras p/ Autoconfig. de Inter. de Senso. e de Disse. (Parte 1) . . .	51
A.2	Regras p/ Autoconfig. de Inter. de Senso. e de Disse. (Parte 2) . . .	51
A.3	Regras p/ Autoconfig. de Inter. de Senso. e de Disse. (Parte 3) . . .	52
A.4	Regras p/ Autoconfig. da Redundância de Informações	52
B.1	Conhecimento sobre Risco de Fogo	54
B.2	Regras para Produção de Contexto de Risco de Fogo	56

Lista de Siglas

AIB	<i>Application Information Base</i>
API	<i>Application Programming Interface</i>
ASP	<i>Answer Set Programming</i>
BC	Base de Conhecimento
BCF	Base de Conhecimento <i>Fuzzy</i>
BDF	Base de Dados <i>Fuzzy</i>
BSN	<i>Body Sensor Network</i>
CPB	<i>Communication Protocol Base</i>
ESA	Elemento Sensor Autônômico
GIS	<i>Geographical Information System</i>
GUI	<i>Graphical User Interface</i>
HVAC	<i>Heating, Ventilation, Ar Conditioning</i>
ISP	<i>In-System Processor</i>
MAPE	Monitoração, Análise, Planejamento, Execução
MIA	Máquina de Inferência Autônômica
MIAD	Máquina de Inferência Autônômica Distribuída
MIAL	Máquina de Inferência Autônômica Local
MIB	<i>Management Information Base</i>
OWL	<i>Ontology Web Language</i>
PAD	Pressão Arterial Diastólica
PAS	Pressão Arterial Sistólica
QoS	<i>Quality of Service</i>
RSSF	Redes de Sensores Sem Fio
TI	Tecnologia da Informação
WSAN	<i>Wireless Sensor and Actuator Network</i>

Capítulo 1

Introdução

Esta dissertação apresenta uma máquina de inferência autonômica distribuída (MIAD) para redes de sensores sem fio (RSSF). Ela aplica lógica *fuzzy* em cada nó sensor de uma RSSF para que estes realizem funções de autoconfiguração de intervalos de sensoriamento, disseminação e redundância de informações da RSSF, assim como, atuem na agregação ou composição distribuída de contexto.

Na Seção 1.1, é apresentada a Inteligência de Ambiente, uma das áreas nas quais a MIAD deve contribuir. Na Seção 1.2, são mostrados problemas na construção de sistemas de monitoração orientados a Inteligência de Ambiente os quais utilizam RSSF como uma das fontes de obtenção de contexto; também são apresentadas as motivações para o desenvolvimento deste trabalho. A Seção 1.3 expõe os principais objetivos desta dissertação e a Seção 1.4 descreve as etapas realizadas para alcançá-los. Por último, na Seção 1.5, é mostrada a estrutura desta dissertação com um pequeno resumo dos capítulos e apêndices.

1.1 Contextualização

Inteligência de Ambiente (*Ambient Intelligence* ou *Smart Environments*) pode ser definida como um ambiente digital que dá suporte a pessoas em suas vidas diárias assistindo-as em modo sensível. Nesse contexto, modo sensível significa que o sistema em questão não só reconhece o usuário, como também aprende e sabe quais suas preferências para assim prover serviços mais personalizados. Logo, Inteligência de Ambiente envolve ambientes enriquecidos com tecnologia e um sistema que toma decisões baseadas em informações obtidas em tempo real e/ou em um histórico

de dados do ambiente, as quais beneficiam os usuários deste lugar. Sensores e dispositivos interconectados em rede fornecem dados para criação dessas informações de apoio ao sistema de decisão (AUGUSTO, 2007).

Portanto, a Inteligência de Ambiente se sobrepõe a outros conceitos, como o da Computação Ubíqua (UbiComp) e Pervasiva, Sensibilidade ao Contexto (*Context Awareness*) e Sistemas Embarcados, porém com algumas diferenças. Sensibilidade ao Contexto pode significar que o sistema tem consciência sobre a situação atual com a qual o usuário está lidando. Este é um dos conceitos mais desejáveis a ser agregado à Inteligência de Ambiente, pois a identificação do contexto é importante para o sistema decidir como atuar de maneira "inteligente" para beneficiar seus usuários (RAMOS, 2007).

1.2 Motivação

Contexto pode ser definido como qualquer informação que pode ser usada para caracterizar a situação de uma entidade. Uma entidade pode ser uma pessoa, lugar ou objeto que é considerado relevante a interação entre um usuário e a aplicação (DEY, 2001). Contextos podem ser obtidos através de dados providos de uma rede de sensores sem fio (RSSF), por exemplo, para os quais são aplicadas técnicas de Inteligência Artificial em sistemas de apoio a decisão centralizados (MÉRICO; MILEO; BISIANI, 2008), (COPETTI; LEITE, 2008).

Segundo Ramos, Augusto e Shapiro (2008), a Inteligência Artificial deve ser combinada às tecnologias de Inteligência de Ambiente (e.g., sensores, sistemas de monitoração do ambiente) para a execução de tarefas de interpretação do estado do ambiente e de tomada de decisões para melhorar a vida de seus usuários. Porém, o sistema de monitoração do ambiente ficará sobrecarregado quanto maior for a quantidade de sensores e a heterogeneidade de informações sensoriadas pela RSSF.

Esse é um problema identificado por Kawashima et al. (2006) e McCarthy (2007). Uma possível solução, ainda pouco explorada, é fazer com que a RSSF funcione como um sistema distribuído e passe a compor outras informações de contexto, a partir de seus dados sensoriados (*raw data*), usando técnicas de inferência, como a lógica *fuzzy* (MARIN-PERIANU; HAVINGA, 2007). Deste modo, aumenta-se o valor semântico das informações de contexto que chegam ao sistema de monitoração do ambiente, pois, parte dessa sobrecarga de pré-processamento será descentralizada para RSSF.

No entanto, devido suas pequenas dimensões, os nós sensores possuem sérias restrições computacionais, como processamento, comunicação e memória, além de poderem ser depositados em lugares de difícil acesso para manutenção. Percebe-se, então, a necessidade das RSSF se tornarem autogerenciáveis, capazes de se manterem e se ajustarem face a mudanças. Para isso, é importante agregar o paradigma da Computação Autônoma às RSSF através do elemento sensor autônomo (ESA), o qual realiza funções de autogerenciamento em nós sensores (BRAGA, 2006).

Nesta dissertação, é proposta uma solução que envolve o uso de técnicas de Inteligência Artificial para produção de contexto distribuída e também para as funções de autogerenciamento das RSSF com a finalidade de contribuir na construção de sistemas de monitoração orientados à Inteligência de Ambiente.

1.3 Objetivo

1.3.1 Objetivo geral

O objetivo deste trabalho é o desenvolvimento de uma máquina de inferência autônoma distribuída (MIAD) para RSSF que utiliza a lógica *fuzzy* em cada nó sensor tanto para realização do autogerenciamento quanto para a composição (ou agregação) distribuída de informações de contexto.

1.3.2 Objetivos específicos

Para atingir objetivo geral deste trabalho, alguns objetivos específicos precisam ser alcançados:

- ▶ modelar conhecimento em conjuntos e regras *fuzzy* para o autogerenciamento de um nó sensor;
- ▶ modelar conhecimento em conjuntos e regras *fuzzy* para produção distribuída de contexto. Isto dependerá do tipo de sensores disponíveis para a implementação da proposta;
- ▶ associar os principais componentes de um sistema de controle *fuzzy* ao modelo do ESA.

1.4 Metodologia

A metodologia empregada na abordagem do problema em questão, bem como na concepção e desenvolvimento da solução proposta nesta dissertação, pode ser resumida como segue.

i. Revisão Bibliográfica

Primeiro, realizaram-se estudos abordando as RSSF sob vários aspectos, como Computação Autônoma, Inteligência Artificial, Inteligência de Ambiente e produção de informações de contexto.

ii. Prática com kits de sensores

Para o desenvolvimento da proposta, foram disponibilizados motes MICAz¹ e placas de sensoriamento MTS400 (temperatura, luz, umidade, pressão e acelerômetro) da empresa Crossbow (2010). Foi estudada a linguagem nesC utilizada na implementação de aplicações para o TinyOS (um sistema operacional para nós sensores de RSSF) (TINYOS, 2010a).

iii. Modelagem de Conhecimento de Autoconfiguração

A partir de regras rígidas (*crisp rules*) de autogerenciamento, foi construída uma base de conhecimento com regras *fuzzy* de autogerenciamento, como também, foram definidas funções de pertinência das variáveis linguísticas utilizadas na autoconfiguração do nó sensor.

iv. Modelagem do Conhecimento de Risco de Fogo

Foi criada uma base de conhecimento com regras *fuzzy* distribuídas (MARIN-PERIANU; HAVINGA, 2007), as quais levam em consideração, as variáveis linguísticas de temperatura e umidade relativa locais e da vizinhança, no processo de inferência de contexto de risco de fogo, feito por cada nó sensor da RSSF. A *baseline* foi construída a partir de uma entrevista com um meteorologista da Fundação Cearense de Meteorologia e Recursos Hídricos - FUNCEME (2010). Nesta etapa houve a realização de vários experimentos envolvendo ar quente e sensores de temperatura e umidade relativa da placa MTS400.

¹placa com unidades de processamento, comunicação e energia

As mensagens de contexto enviadas ao sistema de monitoração do ambiente são baseadas na ontologia μ SONG (HAUIFENG; XINGSHE, 2005), que possibilita os nós sensores da RSSF expressarem de forma simples e flexível o conhecimento de risco de fogo sobre o ambiente monitorado.

v. Implementação e avaliação

Foram desenvolvidos em linguagem nesC e avaliados em uma RSSF com 6 motes MICAz: 1) a MIAD, proposta desta dissertação; 2) o ESA com funções de autogerenciamento baseado em *crisp rules* (BRAGA, 2006); 3) o DFLER, capaz de produzir contexto de forma distribuída, mas sem se preocupar com questões de autogerenciamento (MARIN-PERIANU; HAVINGA, 2007); e 4) uma máquina de inferência autonômica local (MIAL), que utiliza lógica *fuzzy* no autogerenciamento do nó sensor, e que apenas envia *raw data* sensorizados localmente ao sistema de monitoração do ambiente.

1.5 Organização do documento

No Capítulo 2 são descritos os principais componentes da RSSF utilizada na proposta. Em seguida, são apresentados alguns trabalhos sobre o emprego das RSSF em grandes projetos de pesquisa da Inteligência de Ambiente. Também são descritas metodologias de produção de contexto em servidores centralizados e na própria RSSF a partir de técnicas de Inteligência Artificial. São elencados conceitos da Computação Autonômica e como eles são aplicados em RSSF. Por último, nesse capítulo, são apresentados vários trabalhos que usam sistemas *fuzzy* em funções de controle em RSSF.

No Capítulo 3, são descritos os principais componentes da MIAD, proposta desta dissertação, que possibilitam a autoconfiguração de intervalos de sensoriamento, de disseminação e também da redundância de informações dos nós sensores, assim como, a produção de contexto distribuída. No Capítulo 4, são descritas as 4 máquinas de inferência avaliadas, o cenário de testes, bem como, são analisados e comparados 12 parâmetros dessas máquinas implementadas.

Finalmente, no Capítulo 5, fazemos uma síntese das principais conclusões, e contribuições, apontando perspectivas de trabalhos futuros. Nos Apêndices A e B, podem ser encontradas, respectivamente, a base de conhecimento com regras *fuzzy* de autoconfiguração e de risco de fogo.

Revisão Bibliográfica

Este capítulo apresenta a importância e o emprego de RSSF para construção de ambientes inteligentes. Na Seção 2.1, é feita uma descrição dos principais componentes de uma RSSF utilizados na implementação deste trabalho. A Seção 2.2 explica a importância das RSSF para projetos de pesquisa atuais envolvendo Inteligência de Ambiente. A Seção 2.3 mostra exemplos de como produzir informações de contexto em servidores centralizados a partir dos dados sensorizados em uma RSSF. A Seção 2.4 descreve como uma RSSF infere informações de contexto de nível semântico mais alto que seus dados sensorizados. A Seção 2.5 introduz o paradigma da Computação Autônoma e mostra como o mesmo pode trazer benefícios às RSSF. Na Seção 2.6, são apresentados trabalhos que utilizam lógica *fuzzy* em funções de controle nos nós sensores da RSSF. E, por fim, na Seção 2.7, conclui-se com uma discussão de como utilizar a lógica *fuzzy* e a Computação Autônoma em RSSF para diminuir a sobrecarga, em um servidor de contextos.

2.1 RSSF e seus principais componentes

Sohraby, Minoli e Znati (2007) definem uma Rede de Sensores Sem Fio (RSSF) como uma infra-estrutura formada por elementos de sensoriamento (medição), processamento e comunicação que dá ao administrador a habilidade de instrumentação, observação e reação a eventos e fenômenos em um ambiente específico. O administrador é tipicamente uma entidade civil, governamental, comercial ou industrial. O ambiente pode ser o mundo físico, um sistema biológico ou um *framework* de tecnologia da informação. Esse tipo de rede é dependente das

suas aplicações, dentre as quais destacam-se, monitoramento ambiental, controle de automação industrial, vigilância militar e telemetria médica (envio de sinais vitais de pacientes a um sistema de monitoração remoto).

Para o desenvolvimento da proposta desta dissertação, foram disponibilizados kits de sensores da empresa Crossbow (CROSSBOW, 2010), cuja composição básica de um nó sensor consiste em uma placa de aquisição de dados com vários sensores encaixada ao *mote*, que é uma outra placa com componentes de rádio, processamento e energia. A placa de sensores MTS400/420, ilustrada na Figura 2.1(a), oferece 5 tipos básicos de sensores de ambiente através de:

- ▶ chip sensor de umidade relativa (0 a 100%) e de temperatura (-40°C a 80°C);
- ▶ chip sensor de pressão barométrica (300 a 110 mbar) e de temperatura (-10°C a 60°C);
- ▶ chip sensor de luz (400 - 1000nm);
- ▶ acelerômetro de 2 eixos ($\pm 2g$, onde $1g = 9.98 m/s^2$), que pode ser usado para detecção de inclinação, movimento, vibração e/ou medição sísmica;
- ▶ módulo GPS (adicional somente ao modelo MTS420 o qual não faz parte dos kits disponibilizados).

O mote MICAz da Figura 2.1(b), também identificado como MPR2400 (banda ISM 2GHz), usa o Chipcon CC2420¹ como *transceiver* de rádio frequência do ZigBee/IEEE 802.15.4. Ele utiliza um modem banda básica do tipo DSSS (*Digital Spread Sequence Spectrum*) que permite uma taxa de dados efetiva de 250 kbps.

O rádio do MICAz é integrado a um microcontrolador Atmega128L² com memória de programa (memória *flash*) de 128KB e memória SRAM de 4KB. A memória *flash* armazena o código da aplicação do nó sensor enquanto a SRAM é usada para armazenar parâmetros da aplicação do usuário e variáveis do sistema operacional TinyOS. Também possui um conector de expansão de 51 pinos de E/S, onde é encaixada a placa de sensores. A unidade de energia do MICAz funciona com duas pilhas alcalinas.

¹<http://focus.ti.com/docs/prod/folders/print/cc2420.html>

²http://www.atmel.com/dyn/resources/prod_documents/doc2467.pdf

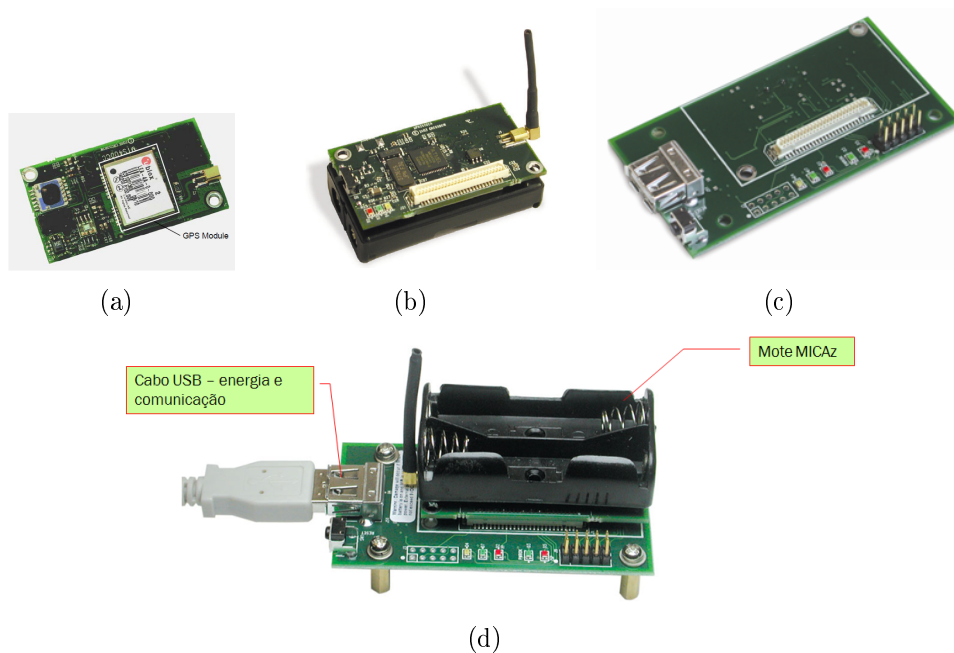


Figura 2.1: (a) MTS400/420 (CROSSBOW, 2007b), (b) mote MICAz, (c) MIB520 e (d) Gateway da RSSF (CROSSBOW, 2007a).

A MIB520 da Figura 2.1(c) provê conectividade USB para programação e comunicação para os motes. Ela fornece energia do computador pela porta USB para os dispositivos conectados, como mostrados na Figura 2.1(d). Ela possui um ISP (*In-System Processor*), responsável por programar o mote com o código trazido do computador pela interface USB. Logo, a MIB520 oferece duas portas via a interface USB: a de programação e a de comunicação. Esta última permite que um mote conectado a MIB520 funcione como *gateway* da RSSF (CROSSBOW, 2007a).

O sistema operacional TinyOS foi criado pela Universidade de Berkeley da Califórnia, EUA, e, hoje, representa um esforço de uma comunidade aberta (*open-community*) para o desenvolvimento de uma plataforma de código aberto (*open-source*) para RSSF. O TinyOS usa como linguagem de programação a nesC (variante do C), cujo compilador reduz o tamanho do código da aplicação na memória do nó sensor.

A linguagem nesC provê uma arquitetura baseada no modelo de componentes, permitindo ao TinyOS possuir uma biblioteca de componentes de protocolos de redes, *drivers* de sensores e ferramentas de aquisição de dados, que podem ser utilizados tal como estão ou customizados para uma aplicação específica. O TinyOS conta com um modelo de concorrência orientado a eventos, que possibilita

agendamento de tarefas (ações) que serão executadas quando ocorrer determinado evento relacionado a comunicação, ao mundo físico ou até mesmo ao funcionamento interno da aplicação (TINYOS, 2010a).

2.2 RSSF e a Inteligência de Ambiente

Dados capturados por nós sensores da RSSF podem ser utilizados na criação de informações de contextos, os quais ajudam na tomada de decisões orientadas às preferências e necessidades de seus usuários. Assim, como ilustrada na Figura 2.2, as RSSF podem ser usadas para produção de contexto de pessoas (pressão arterial, batimentos cardíacos, temperatura) através de Redes de Sensores de Corpo (*Body Sensor Networks* - BSN), contexto do ambiente (luz, temperatura, umidade, pressão), entre outros.

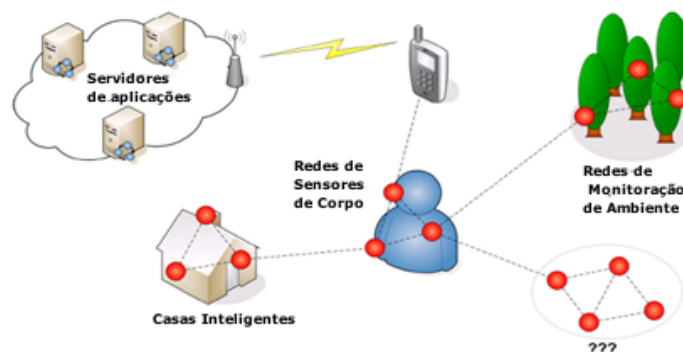


Figura 2.2: Cenário do e-SENSE (ESENSE, 2007).

e-SENSE

O projeto e-SENSE teve o objetivo de capturar Inteligência de Ambiente para comunicações móveis através de RSSF. O projeto viveu de janeiro de 2006 e dezembro de 2007 e foram pesquisadas técnicas para a convergência de informações de contexto de diferentes sensores sem fio para se deduzir o estado emocional atual do usuário, configurações que buscassem o bem-estar social e informações gerais do ambiente monitorado. As aplicações de dispositivos móveis podem acessar um sistema de monitoração que implementa essas técnicas para recuperar as informações de contexto de interesse de forma síncrona (requisições) ou assíncrona (subscrições).

A Figura 2.2 ilustra as informações vindas de RSSF de uma casa inteligente (*smart house*), de uma BSN no corpo de uma pessoa e de vários outros ambientes sendo reunidas e processadas em servidores para provisão de contexto às aplicações

clientes. Por exemplo, se um paciente não pode sair de seu quarto ou casa, um contexto de interesse para seus médicos e enfermeiros seria "paciente saiu do quarto". Caso esse contexto fosse produzido, o servidor o enviaria aos *smartphones* dos funcionários interessados em recebê-los, como também, ao sistema de som do ambiente. Assim, um serviço inteligente seria uma aplicação no *smartphone* chamar a atenção dos funcionários através de alarmes, além de, acionar uma gravação no sistema de som pedindo para o paciente voltar ao seu quarto (ESENSE, 2007).

SENSEI

O consórcio europeu SENSEI continua explorando o objetivo de e-SENSE de maneira mais audaciosa. Composto por 19 parceiros (grandes universidades e empresas) de 11 países europeus, o SENSEI foi iniciado em fevereiro de 2008 e terá duração de 3 anos. De acordo com o SENSEI, o mundo em que vivemos é dividido em um mundo real e um mundo digital.

O mundo real consiste em ambientes físicos instrumentados³ por RSSF e atuadores (*Wireless Sensor and Actuator Networks* - WSN) a fim de se monitorar e interagir com entidades físicas de interesse: pessoas, lugares e objetos. O mundo digital consiste em 1)recursos que são representações de instrumentos (sensores, atuadores, elementos capazes de processar informações); 2)entidades de interesse as quais são representações das pessoas, lugares; e objetos e 3)usuários de recursos que representam as pessoas físicas ou aplicações de software que pretendem interagir com os recursos e as entidades de recursos (Ver Figura 2.3).

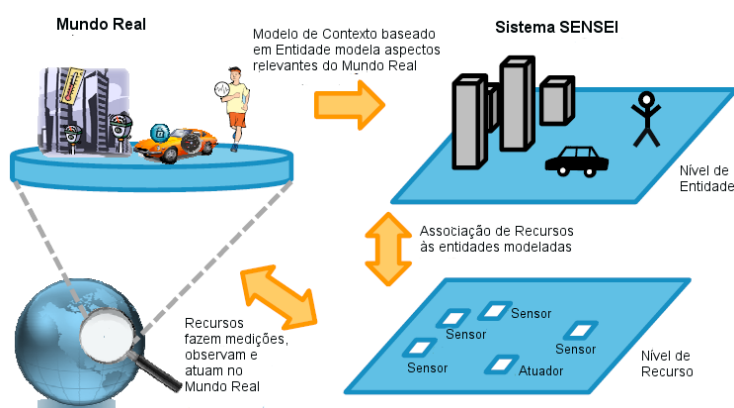


Figura 2.3: Níveis de Abstração do SENSEI (SENSEI, 2010).

³no sentido de instrumentação, ou seja, "resgatar informações que permitam acompanhar o desenrolar de algum processo físico ou químico"(MICHEL, 2009)

A principal contribuição do SENSEI será uma interface que liga o mundo real ao mundo digital através da oferta de dois níveis de abstração mostrados na Figura 2.3 permitindo, desta maneira, a usuários e aplicações interagirem com os Recursos e as Entidades de Interesse. Portanto, o SENSEI tornará possível a criação de cidades inteligentes que proporcionarão desde uma melhor conectividade e interação entre indivíduos e organizações até transportes inteligentes que buscam sempre a melhor rota, enquanto se preocupam em reduzir a emissão de monóxido de carbono na atmosfera de acordo com o nível de poluição do ambiente (SENSEI, 2010).

2.3 RSSF e Sistemas Sensíveis ao Contexto

Como mencionado na Seção 1.1, a Inteligência de Ambiente abrange vários conceitos, dentre eles, o de Sensibilidade ao Contexto (*Context Awareness*).

Arquitetura em Camadas de um Servidor de Contextos

Baldauf, Dustdar e Rosenberg (2004) afirmam que em sistemas sensíveis ao contexto, deve-se separar a detecção do uso do contexto para aumentar a extensibilidade e usabilidade desses sistemas. Por isso, os autores propõem uma arquitetura conceitual de 5 camadas (Figura 2.4), a qual acrescenta, além da detecção e uso, funcionalidades de interpretação e raciocínio de contexto.



Figura 2.4: Arquitetura Conceitual em Camadas (BALDAUF; DUSTDAR; ROSENBERG, 2004).

A primeira camada pode conter sensores físicos, virtuais e lógicos. São exemplos de sensores físicos fotodiodos, câmeras, sensores de toque, acelerômetros. Sensores virtuais fornecem informações de contexto a partir de aplicações ou serviços de software, por exemplo, a partir da análise de movimentos do mouse e da entrada de dados de um teclado, pode-se detectar a atividade atual do usuário. Sensores

lógicos usam várias fontes informações, além de combinar sensores físicos e virtuais com informações de bases de dados para solucionar tarefas complexas.

A segunda camada dessa arquitetura tem a principal função de recuperar os dados de contexto vindos diretamente dos sensores (*raw data*) fazendo uso de *drivers*, para os sensores físicos ou de APIs (*Application Programming Interface*), para os sensores virtuais e lógicos.

A terceira camada é a de pré-processamento responsável pelo raciocínio e interpretação das informações de contexto, pois, *raw data* podem não ser adequados e não ser usados pelos projetistas das aplicações clientes. Nessa camada, se realiza a **agregação** ou **composição** a qual consiste na combinação de dados de contexto de diferentes fontes através de métodos estatísticos ou técnicas de Inteligência Artificial resultando em contextos de nível semântico mais alto que os *raw data*. Nesta dissertação, a partir dos *raw data* de temperatura e umidade relativa, o nó sensor é capaz de produzir o grau de risco de fogo, o qual possui maior valor semântico.

A quarta camada de armazenamento e gerência organiza os dados coletados oferecendo-os às aplicações clientes por meio de interfaces públicas de consultas ao servidor de contexto (forma síncrona) ou subscrições (forma assíncrona). As instâncias dos contextos e reações aos diferentes eventos são implementadas na camada de aplicação. O Modelo aqui descrito é apenas conceitual e deve ser implementado por uma aplicação servidor de contexto, o qual fará interface entre as aplicações clientes e os nós sensores RSSF.

Produção Centralizada de Contexto Médico

Copetti e Leite (2008) apresentam o *framework* H-SAUDE (*Health Support in Aware and Ubiquitous Environments*), o qual oferece suporte ao monitoramento inteligente e sensível ao contexto da saúde de pessoas em casa (*remote assisted living*). No cenário proposto, existe uma BSN no paciente coletando a pressão arterial sistólica (PAS) e a pressão arterial diastólica (PAD) e também sensores de temperatura, acelerômetros e detectores de presença nos principais locais da casa. As RSSF enviam os dados para um computador da casa, onde o H-SAUDE executa. Primeiro, informações de contexto de nível semântico mais alto é produzido a partir dos dados da RSSF. Por exemplo, a partir dos dados de acelerômetros da BSN do paciente, é possível verificar se o paciente está deitado ou em pé. Assim, a informação de contexto "Paciente em pé" tem maior nível semântico que dados numéricos providos pelos acelerômetros.

Segundo, o contexto é encaminhado para um módulo de Raciocínio o qual vai inferir a situação do paciente com o auxílio de regras *fuzzy* tais como: "Se a pessoa está em repouso e a variação da PAS e da PAD é baixa enquanto em repouso então a situação é de alerta". A situação inferida do paciente pode gerar uma ação junto ao sensor, uma interação com o paciente ou lançamento de um alerta para outros usuários do sistema que cuidam do paciente. Nesse trabalho, apenas os dados dos sensores foram disponíveis e o componente de Raciocínio foi implementado em Matlab.

Merico, Mileo e Bisiani (2008) enfatizam o uso de RSSF para coleta de dados do ambiente (luz, temperatura, umidade) em conjunto com um *framework* lógico para representar conhecimento e raciocínio em sistemas domésticos de monitoração da qualidade de vida e prevenção de acidentes. A RSSF utiliza controle de topologia organizando os nós sensores em grupos (*clusters*) delimitados por áreas e salas da casa. O nó líder pode colocar os outros nós de seu *cluster* no modo dormindo (*sleep*) com o objetivo de economizar a energia da RSSF e eliminar a redundância de dados no sistema de monitoração, porém, não foi mencionado os valores desses ganhos, nem o número de nós sensores. Os dados coletados são enviados a um sistema centralizado que utiliza ASP (*Answer Set Programming*) e ontologias para representação de conhecimento e raciocínio das entidades de pessoa, localização, atividade e objeto, bem como, deixar bem definidas os relacionamentos entre essas entidades a fim de se possibilitar a interpretação do contexto relacionados à saúde da pessoa e ao risco de acidentes domésticos.

Redes de Sensores Semânticas

Kawashima et al. (2006) usam o termo Redes de Sensores Semânticas (*Semantic Sensor Networks*) para dizer que é possível descrever o contexto de objetos do mundo físico pela aquisição dinâmica de conhecimento obtido de *raw data* da RSSF. Em seus experimentos, foram utilizados diferentes tipos de sensores sem fio, incluindo motes e placas da família MTS (CROSSBOW, 2010), anexados a diferentes tipos de objetos como garrafas, copos e livros. Os autores não mencionam o número de nós sensores utilizados nem a quantidade de energia gasta por eles nos experimentos.

Em um servidor centralizado, foram instaladas duas bases de dados: a primeira, contendo informações sobre o ambiente físico (dados de sensores e seus metadados) e a segunda, contendo informações sobre nós sensores e o relacionamento destes com os objetos do mundo físico. As bases de dados podem ser acessadas por uma

GUI (*Graphical User Interface*) e por um robô do tipo humanóide. O servidor tem funções de gerenciar os dispositivos sensores, as informações do ambiente e o processamento de consultas às bases de dados.

Depois de receber um dado vindo da RSSF, o servidor gera dinamicamente metadados, anexando ao dado em questão. Então, o relacionamento lógico entre os objetos do mundo físico é determinado através de regras de inferência descritas em linguagem natural. Um exemplo de regra de criação de metadados a partir de dados de um acelerômetro de 3 eixos anexado a um objeto: "Se o eixo Z de um objeto é menor que 100 mm, esse objeto deve ter sido colocado sobre o chão".

Essa regra gera o metadado "Objeto sobre o chão". Um exemplo de regra que infere o relacionamento entre objetos: "Se a bebida está sobre a livro, o livro corre o risco de ser molhado". Um copo ou garrafa cheio de líquido representariam a "bebida" nesta regra. Assim, após a geração dos metadados, o servidor armazena o dado anexado aos seus metadados. Quando um usuário do sistema (pessoa ou robô) acessar essas informações, ele poderia procurar e retirar o objeto que está sobre o chão ou guardaria o livro em local seguro, por exemplo.

Produção de Contexto com dados de Teias de Sensores

Teias de Sensores (*Sensor Webs*) são ferramentas automatizadas de coleta e armazenamento de observações vindas de diversos tipos de sensores, incluindo as de nós sensores de RSSF, que monitoram ambientes como as superfícies e subsolos, oceanos e atmosfera. Elas são consideradas sistemas de informação geográficos (*Geographical Information System - GIS*), pois facilitam a coleta, a distribuição e a disseminação de grandes quantidades dados espaciais significantes. Várias organizações pesquisam formas de fazer as teias de sensores reagirem e adaptarem seu comportamento baseado em fatores externos do ambiente e diagnósticos internos a teia (ALLIANCE, 2010).

McCarthy (2007) usa dados das teias de sensores em monitoração de paisagens, em especial, na monitoração de risco de deslizamento em declives montanhosos. É apresentada uma metodologia para transformar automaticamente os *raw data* dos sensores em representações usadas no processo de raciocínio com ontologias que modelam tanto o sensor como o domínio do problema da detecção de risco.

Os dados vindos dos sensores são armazenados em uma base de dados centralizada do Serviço de Observação de Sensores (SOS) junto a descrições dos

sensores. O SOS começa o processo de transformação codificando registros da sua base de dados em documentos da SensorML (*Sensor XML*) e O&M (*Observation and Measurements*). Em seguida, os documentos SensorML e O&M são convertidos em OWL (*Ontology Web Language*) alinhando-os a ontologia conceitual desenvolvida para o domínio do problema. Depois, os documentos OWL são transformados em uma ontologia operacional em CLIPS (*C Language Integrated Production System*), uma linguagem que lida com fatos e regras utilizada no desenvolvimento da máquina de raciocínio do sistema especialista desse trabalho.

2.4 Inteligência Artificial aplicada a RSSF

Tanto Kawashima et al. (2006) quanto McCarthy (2007) observaram que quanto maior o número de sensores e a diversidade (heterogeneidade) dos dados, maior será o volume de dados concentrados no servidor de contextos, aumentando, desta maneira, a sobrecarga durante agregação ou composição de contextos de nível semântico mais alto do ambiente monitorado. Uma possível solução para este problema seria a própria RSSF tornar-se um sistema distribuído que agregasse os próprios dados sensorizados, através da Inteligência Artificial, para inferir o contexto. Logo, um nó sensor da RSSF pode produzir informações de contexto a partir de dados sensorizados localmente e/ou obtidos de forma colaborativa, trocando informações de contexto com seus nós sensores vizinhos.

Produção Local de Contexto com Regras Rígidas

Marin-Perianu et al. (2006) propõem uma arquitetura de lógica e regras de negócio (*business rules*) para os nós sensores e às aplicações que fazem parte do sistema de monitoração do ambiente. A arquitetura ao nível dos nós sensores foi testada em 5 nós sensores μ Node da empresa Ambient (2010), cada qual com sensores de temperatura, luz e umidade. As regras ao nível do nó sensor compreendem as condições do ambiente expresso pelos dados sensorizados localmente e pelas ações que vão ser tomadas de acordo com o contexto do ambiente.

Por exemplo, para temperatura t e umidade u monitoradas, segue a regra: "Se $t \notin [t_{min}; t_{max}]$ ou $u \notin [u_{min}; u_{max}]$ ou $\Delta t > \Delta t_{max}$ então existe um 'distúrbio de temperatura' no ambiente. Acionar o sistema HVAC". Na regra anterior, a condição especifica os limiares mínimos e máximos dos dados sensorizados ($t_{min}, t_{max}, u_{min}, u_{max}$), o que a caracteriza como uma regra rígida (*crisp rule*). Δt significa a variação do último dado sensorizado e Δt_{max} , a maior variação admissível.

A ação consiste em iniciar o sistema HVAC (*Heating, Ventilation, Ar Conditioning*) responsável pelo aquecimento, ventilação e o ar-condicionado do ambiente.

Foram modeladas 7 regras desse tipo e o código dessa aplicação e os *drivers* dos sensores juntos consumiram 7,3% da memória *flash* do nó sensor. Não foi mencionado a quantidade de memória consumida pelo kernel do sistema operacional e os protocolos de redes, nem a quantidade de energia consumida.

Produção Distribuída de Contexto com Regras *Fuzzy* - DFLER

Marin-Perianu e Havinga (2007) criaram o DFLER, um sistema *fuzzy* distribuído para os nós sensores de RSSF inferirem o contexto de decisão de fogo no ambiente a partir de dados de sensores de temperatura e fumaça locais e da vizinhança. Os dados locais e suas variações são fuzzificados através de conjuntos *fuzzy* pré-determinados e depois enviados aos nós vizinhos via broadcast da camada MAC.

Uma propriedade de interesse de um conjunto *fuzzy* F é caracterizada pela função de pertinência $\mu_F(x)$, a qual dá o grau de semelhança de x a F . Assim, x pode ser o valor do dado sensoriado e sua variação e F pode ser, por exemplo, "nível de fumaça alto". No processo de fuzzificação⁴, uma variável linguística "nível de fumaça alto" é instanciada com o valor de $\mu_F(x)$.

Quando um nó sensor recebe uma observação de um vizinho, ele atualiza o fator contador-sigma (*sigma-count factor*) definido por:

$$\sum Count(F) = \sum_i \mu_F(x_i). \quad (2.1)$$

Onde $X = \{x_1, \dots, x_n\}$ é o conjunto de nós vizinhos.

Depois, as observações são caracterizadas por um quantificador de maioria que dá uma indicação fuzzificada da opinião consensual da vizinhança a qual o nó sensor adiciona as suas observações locais, para, então tomar uma decisão mais exata. Ele é definido pela função *most*:

$$\mu_{most}\left(\frac{\sum Count(F)}{|X|}\right) = \mu_{most}\left(\frac{\sum_i \mu_F(x_i)}{n}\right). \quad (2.2)$$

As regras *fuzzy* do DFLER são "distribuídas", pois, abrangem tanto os dados

⁴"neologismo para traduzir *fuzzification*, pode-se utilizar também 'imprecisificação'"(CRUZ, 2009)

fuzzificados localmente, quanto as observações da vizinhança como segue o exemplo: "SE *Fumaça está Alta E Temperatura está Baixa E most(FumaçaDaVizinhança) é Alta E most(TemperaturaDaVizinhança) é Alta* ENTÃO *DecisãoDeFogo é Alta*". Várias regras podem ser ativadas e são combinadas pela inferência *max-min*. A última fase é a defuzzificação⁵ dos valores que produz saídas rígidas (*crisp*) para o grau da "decisão de fogo" calculada usando o método Centróide.

O DFLEER foi testado no simulador OMNet++ e comparado com outros métodos de inferência como o de 1) regras baseadas em limites bem definidos (*crisp rules*), 2) média dos dados locais e da vizinhança, 3) sistema *fuzzy* local cujas entradas são os dados de temperatura e nível de fumaça sensorizados localmente e suas respectivas variações.

Todos os métodos foram testados com dados que representavam casos reais de fogo e outros dados de falsos positivos, como aqueles vindos de comida queimada, por exemplo. Os casos envolvendo lógica *fuzzy* mostraram ser mais robustos nos cenários com os dados dos falsos positivos, com 0% de decisões equivocadas, sendo menos de 2%, a taxa de erro de média do DFLEER no cenário com os dados de fogo real.

O DFLEER foi também implementado para sensores μ Node da empresa Ambient (2010) e seu código ocupou aproximadamente 1 KB da memória *flash* de 48 KB, 20 bytes da RAM com variáveis estáticas e, durante sua execução, alocou 326 bytes da memória RAM de 10 KB. Nesses valores do consumo da memória, não se incluiu o *kernel* do sistema operacional do sensor, os *drivers* dos sensores e os componentes da pilha de protocolos de redes. Os autores não mencionaram a energia consumida por cada nó sensor.

Produção Distribuída de Contexto - Reconhecimento de Atividades

Em (MARIN-PERIANU et al., 2008), as RSSF reconhecem atividades de funcionários montadores de carros através da lógica *fuzzy* e conhecimento de ordem temporal de sequências de operações envolvidas nessas atividades. Nós sensores sem fio estão presentes nas roupas dos montadores, nas ferramentas e na infraestrutura da montagem. Cada sensor pode produzir uma série de eventos dependendo do seu tipo, os quais serão combinados para se detectar um conjunto de operações básicas.

⁵"neologismo para traduzir *defuzzification*. Pode-se utilizar também 'precisificação'"(CRUZ, 2009)

Por exemplo, a operação "Parafusar uma roda", pode ser inferido por eventos disparados por sensores no braço do montador, na chave de fenda e na parte do carro montada. A partir de uma sequência de operações básicas pode se inferir atividades mais complexas. Por exemplo, "Montar a luz do freio" é composta das operações "Pegar a luz do freio", "Encaixar a luz do freio" e "Parafusar".

São entradas no sistema *fuzzy* as operações básicas da atividade e o valor de um fator que tem a função de punir a ativação da regra quanto mais errada estiver a sequência das operações. Para esse tipo de aplicação, foi utilizado o método de inferência soma-produto e para a defuzzificação, o método maior dos máximos. O sistema foi testado com dados de experimentos de 12 acelerômetros de 3 eixos, os quais descrevem 11 atividades complexas e um total de 49 eventos.

Cada atividade foi executada 10 vezes onde foram analisados a precisão, que indica o quanto o método classifica as condições do ambiente como elas realmente são e a revocação, que se refere a todas as medições específicas de uma condição específica do ambiente indicando somente aqueles que foram classificadas corretamente. Assim, o sistema *fuzzy* com regras sem o fator de punição temporal obteve 0,79 para a precisão e 0,81 para a revocação, enquanto, o sistema *fuzzy* com regras com o fator de punição obteve 0,85 tanto para precisão como para revocação.

O sistema foi implementado para plataforma de nó sensor Tmote e ocupou 1,3 KB de RAM e 1,9 KB de *flash* o que representa um baixo requisito de memória. Mais uma vez, os autores não mostraram a quantidade de energia consumida por cada nó sensor. É importante ressaltar que os trabalhos descritos nesta seção compõe parte da tese de doutorado de Marin-Perianu (2008), além de ter sido suportado pelo projeto SENSEI (2010).

2.5 RSSF e a Computação Autônômica

RSSF possui sérias restrições de hardware, comunicação e software e há necessidade de intervenção humana na troca de componentes dos nós sensores, por exemplo, baterias, para que esta se mantenha operacional por muito tempo. Porém, muitas vezes, essas redes são empregadas em regiões de difícil acesso humano, o que não torna viável essa manutenção. Portanto, é essencial que RSSF se autogerenciem a fim de maximizar as atividades de suas aplicações incorporando o paradigma da Computação Autônômica.

2.5.1 Computação Autônômica

Computação Autônômica incentiva a construção de sistemas computacionais, hardware e software, que gerenciam a si próprios. Esse termo foi introduzido por Paul Horn, vice-presidente sênior de pesquisa da IBM em 2001 e é baseado no sistema nervoso autônômico humano. Por exemplo, os batimentos cardíacos são ações realizadas pelo corpo humano sem que o homem expresse voluntariamente esses movimentos. O funcionamento de órgãos internos como o aparelho digestivo, respiratório ou renovação da epiderme não depende da vontade humana, funcionam de forma totalmente transparente a esta (HORN, 2001).

O autogerenciamento é a essência da Computação Autônômica, e seu intuito principal é liberar os administradores de sistemas de detalhes de operação e manutenção e prover aos usuários sistemas computacionais capazes de trabalhar com alto desempenho 24 horas por dia e 7 dias por semana. Como nos sistemas biológicos, os sistemas da Computação Autônômica irão manter e ajustar sua operação face às mudanças de componentes, cargas de trabalho, demandas e condições externas e também face às falhas maliciosas, ou não, no seu hardware ou software (KEPHART; CHESS, 2003). Ou seja, aos usuários restarão tarefas de alto nível e ao sistema computacional, decisões e ações de baixo nível. Uma das características de sistemas autogerenciados é a autoconfiguração.

A autoconfiguração menciona que o sistema não precisa esperar para que seu operador o configure ou reconfigure manualmente, pois, ele mesmo tomará a iniciativa de executar essas ações quando achar necessário. Por exemplo, quando um novo componente for introduzido ao sistema, ele deverá automaticamente aprender e levar em consideração a composição e configuração do sistema. E ele deverá informar ao sistema sobre suas funções para que este possa utilizá-lo.

2.5.2 RSSF Autônômicas

Segundo (KEPHART; CHESS, 2003), um sistema autônômico consiste em vários elementos autônômicos que interagem entre si. Elementos autônômicos, por sua vez, são sistemas individuais que contêm recursos e serviços prestados a humanos ou a outros elementos autônômicos. Eles devem gerenciar o seu próprio comportamento interno e os relacionamentos com outros elementos autônômicos de acordo com políticas estabelecidas por humanos ou outros elementos autônômicos. A Figura 2.5 mostra um elemento autônômico composto por um ou mais elementos gerenciados

e um gerente autônomo que o controla e o representa, ou seja, ele interage com outros elementos autônômicos e humanos através de seu gerente autônomo.

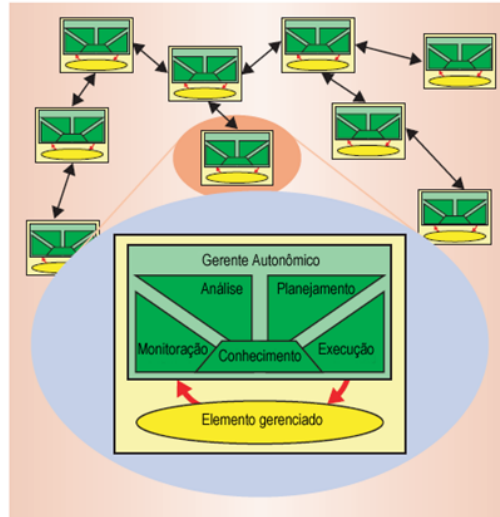


Figura 2.5: Estrutura de um Elemento Autônomo (KEPHART; CHESS, 2003)

O Serviço de Monitoração provê as características de autoconhecimento, a qual este deve ter conhecimento sobre si mesmo, e de autoconsciência, a qual este deve conhecer também o ambiente que o cerca. O desenvolvimento destas características dependerá do tipo de tecnologia que a rede autônoma utiliza, pois, isso envolve vários parâmetros de configuração de hardware, software, e do ambiente que serão monitorados.

O Serviço de Análise implementa uma outra característica autônoma, o autodiagnóstico, tratando os dados recolhidos pelo Serviço de Monitoração do elemento gerenciado de modo a obter informações úteis para análise e tomada de decisões, se apoiando também na sua Base de Conhecimento. O produto desse serviço será o diagnóstico sobre o elemento autônomo e sua vizinhança que pode desencadear uma série de ações a serem executadas para manter o sistema otimizado, protegido e livre de falhas.

O Serviço de Planejamento irá processar o diagnóstico feito pelo serviço de análise e irá definir quais ações serão tomadas. Aqui podem ser desenvolvidas outras características autônomas como o autoserviço, autosustento, auto-organização e etc, que dependerão do grau de precisão e nível de maturidade autônoma, como também da complexidade permitida. Técnicas de Inteligência Artificial e modelos estatísticos podem ser utilizados se o elemento autônomo tiver recursos

computacionais para isso.

O Serviço de Execução é marcado pela autoconfiguração, pois este irá configurar o hardware e software envolvido, buscando satisfazer às necessidades do elemento autônômico e de seu ambiente da melhor forma possível. Ele apenas executará o que foi estabelecido pelo Serviço de Planejamento, sem se preocupar se existem *bugs* em tais ordens.

ESA

(BRAGA, 2006) propõe um Elemento Sensor Autônômico (ESA) para RSSF baseado no modelo genérico de (KEPHART; CHESS, 2003). A Figura 2.6 apresenta uma visão geral do funcionamento do ESA. Ele é composto por: Serviço de Monitoração, Análise, Planejamento, Execução, Base de Conhecimento e Elemento Gerenciado.

A base de conhecimento (BC) é composta por uma MIB (*Management Information Base*), uma AIB (*Application Information Base*), uma máquina de políticas e informações sobre o protocolo de comunicação utilizado na rede. A MIB armazena informações coletadas ou produzidas e utilizadas pelos serviços de monitoração, análise, planejamento e execução do ESA, como características da rede, dos elementos da rede, serviços ou negócio.

AIB deve conter apenas limites e representações de dados, tais, como mapas específicos da aplicação, pois, é importante que a entidade de gerenciamento, conheça e utilize detalhes gerenciados. A máquina de políticas é responsável por armazenar e gerenciar políticas de negócio especificadas pelo projetista da rede. Ela pode ser usada por todos os serviços do ESA, mas em particular pelo serviço de planejamento para definição das ações necessárias, de acordo com os resultados do serviço de análise.

O Serviço de Monitoração monitora parâmetros de hardware e software do nó sensor e os contabiliza. No Serviço de Análise, as informações mencionadas anteriormente são convertidas em eventos, utilizando valores 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 dependendo do tipo parâmetro e armazenados na BC. Os eventos são utilizados em algoritmo de correlação baseado em regras rígidas (SE-ENTÃO), o qual apresentará conclusões sobre o desempenho do nó sensor, ou seja, verificará o nível de QoS (auto

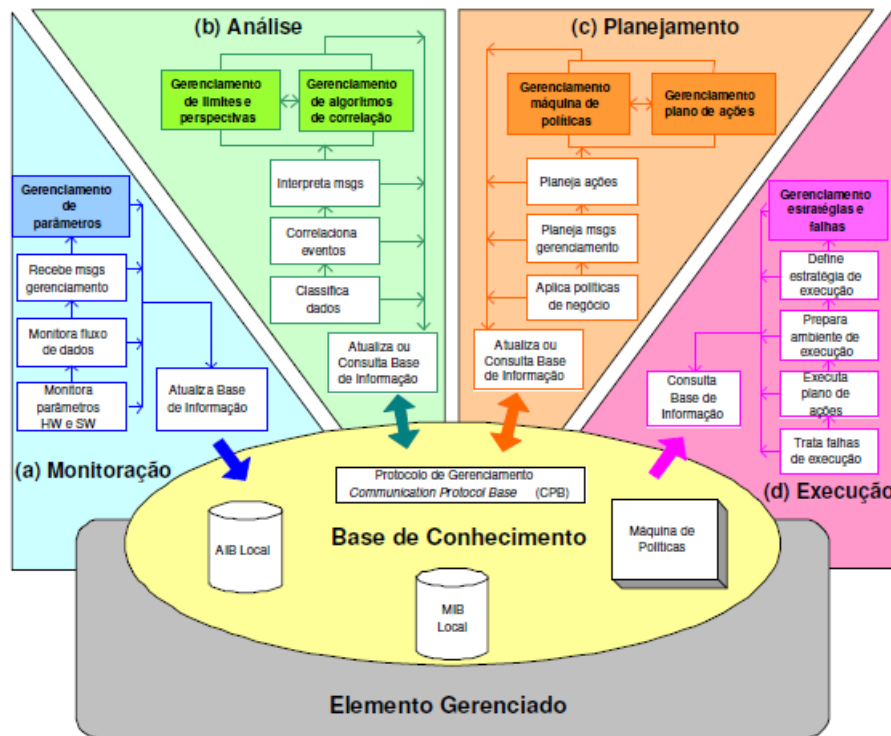


Figura 2.6: Visão Geral dos serviços, funções e componentes de conhecimento que compõem o Elemento Sensor Autônomo (BRAGA, 2006).

diagnóstico).

No Serviço de Planejamento, o plano de ações é elaborado, as quais se restringem ao aumento ou diminuição dos intervalos de sensoriamento e disseminação, bem como em ativar e desativar o nó sensor que possui informações redundantes em determinada região de sensoriamento, mediante o acréscimo de probabilidade de execução destas tarefas. O Serviço de Execução recebe o plano de ações a serem executadas (inclusive o envio de mensagens) e o realiza de acordo com a melhor estratégia de execução possível (autoconfiguração).

Os ESAs da RSSF são iniciados com determinados valores para a taxa de sensoriamento e comunicação considerados médios e normais para as aplicações. A cada 200 segundos, um nó sensor envia mensagens de gerência aos seus vizinhos divulgando as configurações atuais do nó, a saber: dados relevantes sensoriados, intervalos de disseminação e sensoriamento, número total de bytes enviados e energia residual. O ciclo autônomo consiste na execução encadeada dos serviços do motor MAPE (Monitoração, Análise, Planejamento e Execução) em conjunto da Base de Conhecimento e seu intervalo é de 100 segundos.

O ESA foi instanciado para o simulador NS-2 tendo como Elemento Gerenciado uma aplicação de sensoriamento de monóxido de carbono (CO) presente em nós e sensores com especificações semelhantes aos motes MICA2 da (CROSSBOW, 2010). As RSSF Autônomicas Distribuídas foram comparadas às RSSF não-gerenciadas e RSSF Autônomicas Locais. As RSSF não-gerenciadas apenas fazem o sensoriamento e disseminação de dados ao sistema de monitoração. RSSF Autônomicas Locais realizam a autoconfiguração do nó sensor baseado somente em informações locais, ou seja, não enviam mensagens de gerência, além de possuírem como elemento gerenciado, uma aplicação simples de sensoriamento e disseminação de dados. Já as RSSF Autônomicas Distribuídas enviam mensagens de gerência aos nós vizinhos divulgando as configurações para usarem no processo de autogerenciamento.

Os resultados mostraram que as RSSF Distribuídas economizaram cerca de 40% da energia da RSSF, pois enviaram um número de mensagens ao sistema de monitoração menor e também ocasionaram menos colisões e perda de pacotes. Braga (2006) não testou o ESA desenvolvido para o NS-2 em uma RSSF com nós sensores reais.

2.6 RSSF e Sistemas de Controle *Fuzzy*

Marin-Perianu e Havinga (2007) mostrou que a lógica *fuzzy* é uma técnica leve (*soft*) boa para ser usada em sistemas com recursos computacionais limitados como os nós sensores da RSSF. Sistemas *fuzzy* podem lidar com decisões errôneas tomadas a partir de dados de sensores defeituosos ou imprecisos, pois, os valores específicos dos dados sensorizados são transformados em valores qualitativos. Essa técnica também pode reduzir o tempo de implementação e o esforço computacional quando comparado com o cálculo bayesiano, que precisa de grandes quantidades de dados utilizados em métodos estatísticos para se obter as probabilidades a priori.

A lógica *fuzzy* além de ser usada em agregação de informações de contexto pode ser empregada em funções de controle nos nós sensores da RSSF. Os trabalhos apresentados nesta seção foram apenas simulados, pois, seus autores não realizaram testes em RSSF com nós sensores reais, tais como os da empresa Crossbow (2010) e Ambient (2010).

Gomes et al. (2010) apresentam uma extensão do protocolo de roteamento AODV (*Ad hoc On-demand Distance Vector*) que utiliza, como métrica de roteamento, um parâmetro de custo *fuzzy* (*Fuzzy Cost - FC*) calculado a partir da

percentagem de energia restante do nó e do grau de adjacência (número de vizinhos do nó). A rota escolhida será aquela de menor FC, caracterizada por nós com grande adjacência e de maior energia restante. Foram simulados no NS-2 o AODV e sua extensão, o AODV-FWSN. Resultados mostram que o AODV-FWSN aumenta mais a vida útil da RSSF, pois, distribui melhor o tráfego entre os nós sensores mais adequados.

Em (ACHARJEE; PATEL, 2008), ao invés de atribuir um atraso aleatório (*backoff*) na camada MAC após a queda de um link ou falha no próximo nó sensor do caminho, é utilizado um sistema *fuzzy* para o cálculo desse atraso em uma RSSF de monitoração de estruturas metálicas. São levados em consideração a função do nó do próximo pulo (nó sensor e roteador de mensagens ou apenas nó roteador), tipo de rota (primária ou secundária) e quantidade de nós do *cluster*. Os resultados simulados mostram que os nós voltam a se comunicar mais rápido, após falhas na comunicação, do que o método tradicional.

Azim e Jamalipour (2006) calculam um fator de custo para a escolha do nó do próximo pulo (*next hop*) durante o roteamento através de um sistema *fuzzy*, cujas entradas são a distância do nó vizinho candidato, a distância a qual o nó candidato está do caminho mais curto (*shortest path*), a percentagem da energia consumida do nó que está decidindo e se o caminho pelo nó candidato é muito utilizado para o encaminhamento de mensagens. Esse sistema *fuzzy* aumenta a vida útil dos nós sensores segundo resultados simulados no MATLAB.

Um controlador *fuzzy* é utilizado em (XIA; ZHAO; TIAN, 2007) para adaptar o intervalo de disseminação de dados à taxa de perda associada à transmissão de dados de um sensor para um atuador em uma WSN. A taxa de perdas sempre fica a um nível desejado pré-determinado garantindo a QoS desejada para aquela rede de acordo com as simulações feitas.

2.7 Conclusão

Neste capítulo foram apresentados trabalhos mostrando a importância da RSSF na construção de Ambientes Inteligentes. Os dados sensorizados pela RSSF são usados na produção de informações de contexto que vão auxiliar o sistema de monitoração do ambiente a tomar decisões inteligentes que garantam a satisfação de seus usuários. Porém, quanto maior o número de nós sensores da RSSF e a heterogeneidade de seus dados, maior o volume de dados a ser correlacionado

durante a composição de informações de contexto. Assim, aumenta-se a sobrecarga no sistema devido a esse processamento.

Marin-Perianu e Havinga (2007) mostraram que a lógica *fuzzy* pode ser usada pelo nó sensor para composição de contexto trazendo funções da camada de processamento da arquitetura centralizada apresentada por Baldauf, Dustdar e Rosenberg (2004) a fim de que a RSSF as execute de maneira distribuída. Isso pode diminuir a sobrecarga no servidor de contextos da Seção 2.3, pois, parte desta composição de contextos está sendo feito pela RSSF.

A Computação Autônoma pode ajudar ainda mais a aliviar essa situação. Foi visto que o ESA, na Seção 2.5.2, proporciona a autoconfiguração dos intervalos de sensoriamento e disseminação e também a ativação e desativação dos sensores, pois utiliza parâmetros internos e externos ao nó sensor, bem como limites bem definidos desses parâmetros em regras rígidas usadas durante o autogerenciamento. Isso diminuiu o envio das mensagens de dados, a quantidade de perdas e colisões, economizando, deste modo, a energia da RSSF.

Na Seção 2.6, foram mostrados trabalhos que aplicam a lógica *fuzzy* em funções de controle internos de nós sensores da RSSF com bom êxito, embora, os resultados tenham sido obtidos através de simulações. Logo, essa técnica pode ser usada nas funções de autogerenciamento de um motor autônomo MAPE, semelhante ao do ESA.

A proposta deste trabalho visa diminuir a sobrecarga no sistema de Inteligência de Ambiente unindo a agregação de contexto descentralizada às funções de autogerenciamento em nós sensores de uma RSSF através do emprego da lógica *fuzzy*. Vale ressaltar que kits de sensores sem fio descritos na Seção 2.1 foram disponibilizados para a implementação da proposta.

Capítulo 3

Proposta: MIAD

Este trabalho apresenta a proposta de uma máquina de inferência autônoma distribuída (MIAD) que utiliza lógica *fuzzy* na provisão de autogerenciamento em RSSF e na agregação ou composição distribuída de informações de contexto feita pela própria RSSF. Na Seção 3.1, são descritos superficialmente os principais componentes da MIAD. A Seção 3.2 apresenta com detalhes o procedimento de inferência *fuzzy* da autoconfiguração e a Seção 3.3, o procedimento de inferência de contexto de risco de fogo do ambiente, ambos realizados pelos nós sensores da RSSF. Na Seção 3.4, são feitas algumas considerações sobre este capítulo.

3.1 Descrição Geral da MIAD

A estrutura da MIAD se baseia no elemento autônomo de Kephart e Chess (2003) e instância do elemento sensor autônomo (ESA) de Braga (2006). Ela pode ser visualizada em detalhes na Figura 3.1. A MIAD possui dois principais componentes um gerente autônomo e um elemento gerenciado.

O gerente autônomo é responsável pela autoconfiguração do nó sensor através de serviços com funções bem definidos pela Computação Autônoma (ver Seção 2.5.2). Para isso, o gerente autônomo conta com um sistema de controle *fuzzy* o qual irá inferir qual o melhor intervalo de sensoriamento e disseminação de informações de contexto, como também se estas informações são redundantes ao processamento de contexto no sistema de monitoração do ambiente. Por isso, este trabalho denomina o gerente autônomo da MIAD de máquina de inferência autônoma (MIA).

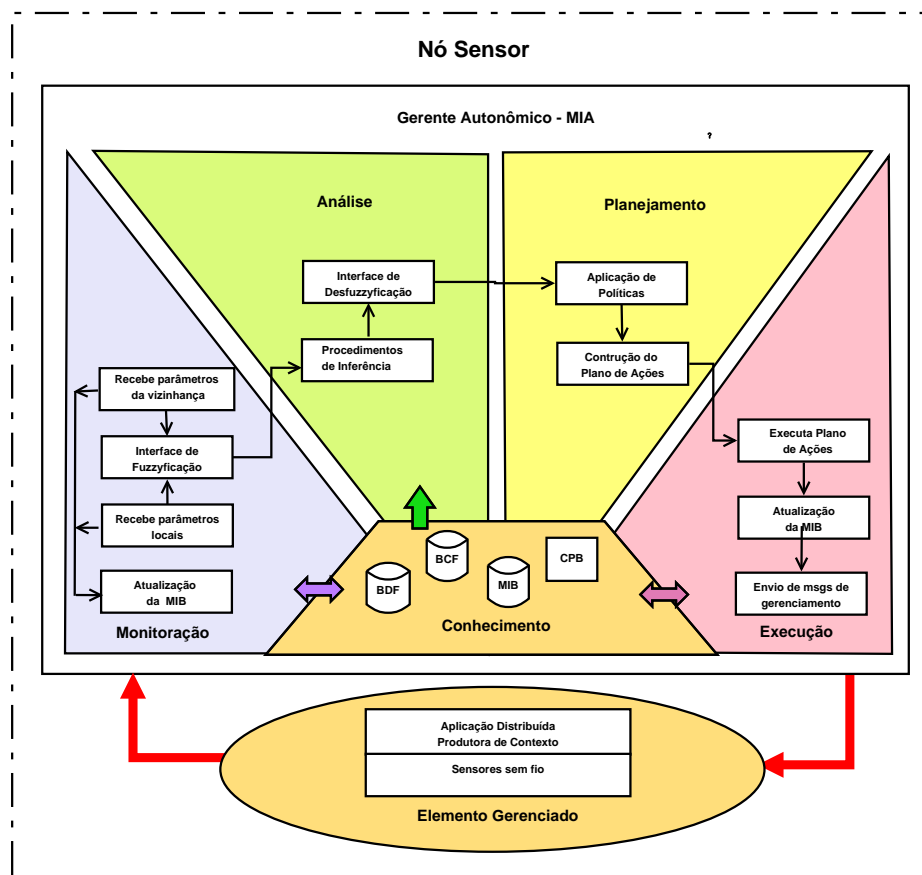


Figura 3.1: Máquina de Inferência Autônômica Distribuída - MIAD

O Elemento Gerenciado da proposta consiste em uma aplicação distribuída produtora de contexto semelhante ao DFLEER (MARIN-PERIANU; HAVINGA, 2007). Ela utiliza dados de temperatura e umidade relativa obtidos por nós sensores compostos de placas MTS400 conectadas a motes MICAz, já apresentados na Seção 2.1, para a inferência do grau de risco de fogo do ambiente monitorado.

3.2 Gerente Autônômico - MIA

A MIA consiste na associação entre um motor autônômico MAPE (Monitoração, Análise, Planejamento, Execução) e Conhecimento e os componentes básicos de um sistema de controle *fuzzy*. A Figura 3.1 mostra como os componentes básicos do sistema controlador *fuzzy* foram englobados pela estrutura da MIAD:

- ▶ Serviço de Monitoração possui a interface de fuzzyficação;
- ▶ Serviço de Análise possui os procedimentos de inferência e interface de desfuzzyficação;

- Conhecimento possui a Base de Dados *Fuzzy* (BDF) e a Base de Conhecimento *Fuzzy* (BCF).

Na MIB (*Management Information Base*) do componente de Conhecimento, ficam armazenados os valores atuais dos parâmetros usados na autoconfiguração do nó sensor, bem como, os limites dos intervalos de sensoriamento e disseminação. A CPB (*Communication Protocol Base*) coleciona o formato das mensagens utilizadas pela MIAD. Nas subseções seguintes, são descritos cada um dos serviços.

3.2.1 Serviço de Monitoração

Conforme ilustrado na Figura 3.1, o Serviço de Monitoração recebe e armazena na MIB o fluxo de parâmetros locais que caracteriza o estado do nó sensor, a seguir: percentagem de dados relevantes, intervalo de sensoriamento, intervalo de disseminação, número de bytes enviados e a percentagem da energia residual do nó sensor. Neste trabalho, um dado de temperatura é considerada relevante quando é maior ou igual a 30 °C e o dado de umidade relativa, quando é menor ou igual a 40% que indicam um grande aquecimento do ambiente (FUNCEME, 2010).

A cada intervalo de gerência, o nó sensor envia esse fluxo de parâmetros locais e os últimos dados sensorizados aos seus nós vizinhos em mensagens de gerência para que o Serviço de Monitoração calcule e armazene na MIB a percentagem de dados relevantes da vizinhança e a percentagem de vizinhos de configuração semelhante. A configuração de um nó sensor vizinho é classificada como semelhante à configuração local quando 60% das informações recebidas através da mensagem de gerência diferem em 20% das respectivas informações disponíveis localmente. Esta é uma convenção utilizada no ESA de Braga (2006). Portanto, a autoconfiguração do nó sensor é baseada tanto em informações locais do nó sensor como em informações dos nós vizinhos, que estão à distância de um *hop* (pulo).

Com o início do ciclo autônômico, a interface de fuzzificação presente no Serviço de Monitoração recebe como entrada os parâmetros locais e da vizinhança armazenados na MIB. A interface de fuzzificação transforma o valor numérico de cada entrada em instâncias (conjunto de termos linguísticos - palavras) da respectiva variável linguística. Atribui-se, então, um grau de pertinência a cada termo linguístico, pela associação do termo a uma função de pertinência do conjunto *fuzzy* da variável linguística em questão (GOMIDE; GUDWIN, 1994).

Deste modo, o valor numérico:

- ▶ da percentagem de dados relevantes e da percentagem de energia residual assumirão quaisquer membros do conjunto *fuzzy* {baixa, média, alta} da Figura 3.2(a);
- ▶ do número de bytes enviados assumirá quaisquer membros do conjunto *fuzzy* {baixo, médio, alto} definidos na Figura 3.2(b);
- ▶ do intervalo de sensoriamento e do intervalo de disseminação assumirão quaisquer membros do conjunto *fuzzy* {baixo, médio, alto} respectivamente as Figuras 3.3(a) e 3.3(b);
- ▶ da percentagem de vizinhos de configuração semelhante assumirá quaisquer membros do conjunto *fuzzy* {baixo, médio, alto} da Figura 3.4.

As funções trapezoidais foram escolhidas porque possuem custo computacional menor que as sinodais, são fáceis de se implementar se adequando as restrições de processamento dos nós sensores.

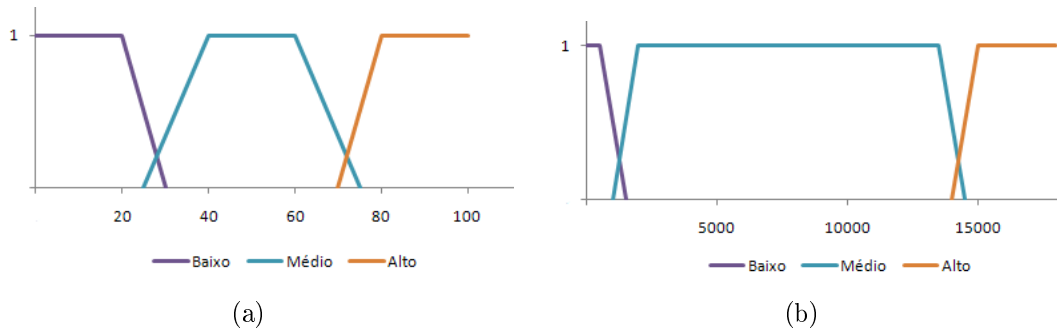


Figura 3.2: (a) Conjunto *fuzzy* da percentagem de dados relevantes e da percentagem de energia residual e (b) Conjunto *fuzzy* do número de bytes transmitidos

Como visualizado nas Figuras 3.2, 3.3 e 3.4, o grau/fator de pertinência quantifica a pertinência de um valor numérico de entrada a um membro de um conjunto *fuzzy* através de um valor entre 0 e 1, onde 0 representa sua completa exclusão do subconjunto e 1, sua máxima pertinência. Assim, um valor numérico de entrada pode ter um grau de pertinência maior, a determinado termo linguístico, do que de outro termo no mesmo conjunto *fuzzy*. Na estrutura da MIAD, Figura 3.1, as definições das funções de pertinência estão localizadas na BDF do componente de Conhecimento.

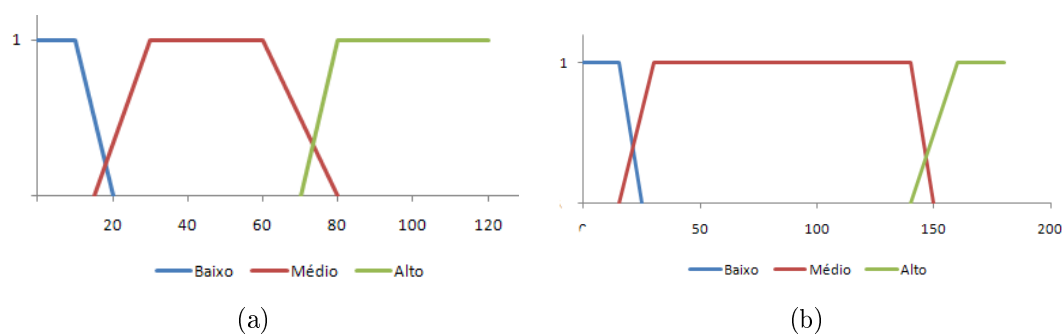


Figura 3.3: (a) Conjunto *fuzzy* do intervalo de sensoriamento e (b) de disseminação

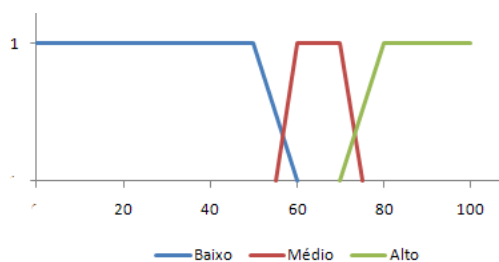


Figura 3.4: Conjunto *fuzzy* da porcentagem de vizinhos de configuração semelhante e da autoconfiguração da redundância de dados

3.2.2 Serviço de Análise

Durante o ciclo autônômico, os parâmetros fuzzificados no Serviço de Monitoração são passados ao Serviço de Análise para que sejam utilizados nos procedimentos de inferência baseados em 74 regras *fuzzy* da BCF do Conhecimento (ver Apêndice A). Estas regras *fuzzy* foram criadas a partir de regras rígidas (*crisp rules*) do trabalho de Braga (2006) e são do tipo "SE-ENTÃO" representando o conhecimento *fuzzy* sobre o estado para autoconfiguração do intervalo de sensoriamento (Figura 3.5(a)), do intervalo de disseminação (Figura 3.5(b)) e da redundância de informações (Figura 3.4), as quais são as variáveis linguísticas.

São exemplos de regras *fuzzy* de autoconfiguração:

- ▶ Regra 41 - Se porcentagem de dados relevantes locais e da vizinhança estão baixas e o intervalo de sensoriamento está alto e o intervalo de disseminação está médio então a configuração do intervalo de sensoriamento e de disseminação é baixa;
- ▶ Regra 68 - Se a porcentagem de vizinhos com mesma configuração é alta e a porcentagem de dados relevantes é baixa então a redundância de informações

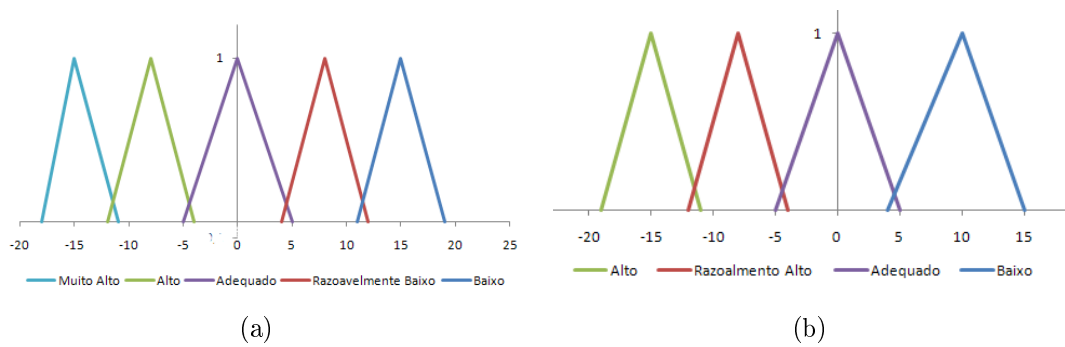


Figura 3.5: (a) Conjunto *fuzzy* da autoconfiguração do intervalo de sensoriamento e (b) de disseminação

é alta.

Em todas as regras, o operador "e" foi utilizado para fazer a interseção (operação *min*) entre os conjuntos *fuzzy* das variáveis linguísticas de entrada. O modelo de inferência utilizado é o Mamdani, onde, neste trabalho, a regra é ativada quando o resultado da operação "e" for maior que zero. Várias regras podem ser ativadas durante o processo de inferência dando cada uma sua contribuição para expressar o estado para autoconfiguração dos intervalos e da redundância. Se isto acontecer, neste trabalho, as contribuições são combinadas pelo operador de união utilizando a operação *max* (GOMIDE; GUDWIN, 1994).

A interface de desfuzzificação recebe o resultado *fuzzy* da inferência e transforma em ações de controle não-*fuzzy* que melhor representa esse resultado através do método de centro de área. No caso dos intervalos, o resultado da desfuzzificação é a quantidade de segundos a ser somada ou subtraída ao valor atual do intervalo (Figura 3.5) e, no caso da redundância, é a porcentagem de quanto a configuração e os dados sensorizados não relevantes de um nó sensor são semelhantes a de seus vizinhos na RSSF (Figura 3.4).

3.2.3 Serviço de Planejamento

As informações desfuzzificadas são encaminhadas ao Serviço de Planejamento, onde são aplicadas políticas de execução da autoconfiguração. Essas políticas envolvem a adição de probabilidade ao aumento ou diminuição dos intervalos e da probabilidade de parar ou não o envio de informações redundantes ao sistema de monitoração do ambiente. Segundo Braga (2006), a probabilidade foi incluída para evitar que todos os nós de uma mesma região diminuam de uma só vez suas taxas

de sensoriamento e/ou disseminação, deixando tal região inteiramente descoberta ou então aumentem suas taxas ao mesmo tempo, reduzindo, deste modo, a possibilidade de colisões e perdas de mensagens.

Seguem as políticas de autoconfiguração definidas neste trabalho para o Serviço de Planejamento da MIAD:

- i. Adicionar probabilidade de 80% de execução para o acréscimo de 12 a 19 segundos do intervalo sensoriamento atual;
- ii. Adicionar probabilidade de 20% de execução para o acréscimo de 4 a 11 segundos do intervalo sensoriamento atual;
- iii. Adicionar probabilidade de 80% de execução para a subtração de 4 a 11 segundos do intervalo sensoriamento atual;
- iv. Executar diretamente a subtração de 12 a 18 segundos do intervalo sensoriamento atual;
- v. Adicionar probabilidade de 80% de execução para o acréscimo de 5 a 15 segundos do intervalo de disseminação atual;
- vi. Adicionar probabilidade de 80% de execução para a subtração de 12 a 19 segundos do intervalo de disseminação atual;
- vii. Adicionar probabilidade de 20% de execução para subtração de 4 a 11 segundos do intervalo de disseminação;
- viii. Adicionar a probabilidade de 40% de execução de parar o envio de mensagens ao sistema de monitoração do ambiente quando a percentagem de redundância de informações for de 55 a 75%;
- ix. Adicionar a probabilidade de 60% de execução de parar o envio de mensagens ao sistema de monitoração do ambiente quando a percentagem de redundância de informações estiver acima de 75%.

Se a quantidade de segundos de algum dos intervalos e/ou a percentagem de redundância (resultados da desfuzzificação) não foi citado pelas políticas definidas, isso indica que não ocorrerá nenhuma ação de autoconfiguração, ou seja, o intervalo em questão continuará o mesmo e/ou o nó sensor não irá parar o envio de mensagens

ao sistema de monitoração do ambiente. Assim, após a aplicação das políticas, é construído um plano de ações de autoconfiguração.

A Tabela 3.1 mostra um exemplo de um plano de ações. Quanto aos intervalos, o valor positivo da defuzzificação representa a ação "adicionar segundos" e o valor negativo, "subtrair segundos". Como explicado nesta seção, as políticas adicionam a probabilidade de execução da ação inferida. Assim, a política i foi aplicada ao intervalo de sensoriamento do exemplo e a política vii, foi aplicada ao intervalo de disseminação. Quanto a redundância de informações, nenhuma política se aplica ao valor desfuzzificado (50), assim, nenhuma ação será executada e o nó sensor continuará enviando mensagens ao sistema de monitoração do ambiente.

Tabela 3.1: Exemplo de um plano de ações

Parâmetro da autoconfiguração	Valor da desfuzzificação	Ação	Probabilidade de Execução
Intervalo de sensoriamento	+13	Adicionar segundos	80%
Intervalo de disseminação	-7	Subtrair segundos	20%
Redundância de informações	50	Nenhuma ação	100%

3.2.4 Serviço de Execução

O plano de ações é executado por este serviço durante o ciclo autonômico. Os intervalos são atualizados apenas se o novo valor não ultrapassar os limites mínimos e máximos estabelecidos na MIB. A saber, para o intervalo de sensoriamento esses limites são respectivamente 10 e 120 segundos e para o intervalo de disseminação, respectivamente, 15 e 180 segundos. O novo valor do intervalo é armazenado na MIB do componente de Conhecimento da MIAD.

3.3 Elemento Gerenciado

O contexto produzido por um nó sensor da RSSF depende dos dados de diferentes sensores. Foram disponibilizadas para a realização destes trabalho placas MTS400 (CROSSBOW, 2010) com sensores para monitoração de temperatura, umidade relativa, luz, pressão e acelerômetros descritos na Seção 2.1. Assim, procurou-se uma base de conhecimento que relacionasse pelo menos dois tipos de dados dessa placa de sensores a fim de se classificar a situação do ambiente.

Não encontrando tal base, foi entrevistado um meteorologista da Fundação Cearense de Meteorologia e Recursos Hídricos - FUNCEME (2010). Além disso, também foram realizados experimentos com os sensores de temperatura e umidade

relativa, porque é possível fazer a agregação ou composição com esses dois tipos de dados para se obter informações sobre o risco de fogo no ambiente. Logo, montou-se uma base de conhecimento com 92 regras *fuzzy* sobre o risco de fogo (ver Apêndice B).

Primeiro, a cada intervalo de sensoriamento, os dados de temperatura e umidade relativa do nó sensor são entradas de um motor *fuzzy* de inferência distribuída semelhante ao DFLER (MARIN-PERIANU; HAVINGA, 2007) para serem fuzzificados de acordo com os conjuntos *fuzzy* das Figura 3.6(a) e 3.6(b), respectivamente. Segundo, os dados fuzzificados são enviados aos nós vizinhos para que seja calculado o quantificador de maioria através das equações 2.1 e 2.2, onde, o quantificador *most* está definido pelas funções da Figura 3.7(a).

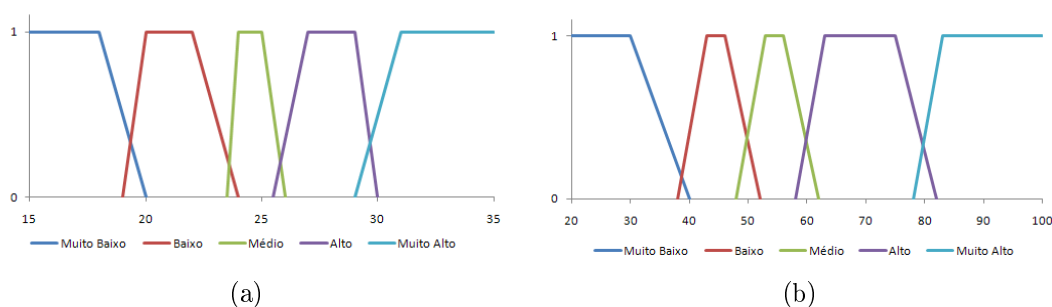


Figura 3.6: (a) Conjunto *fuzzy* de temperatura e (b) da umidade relativa

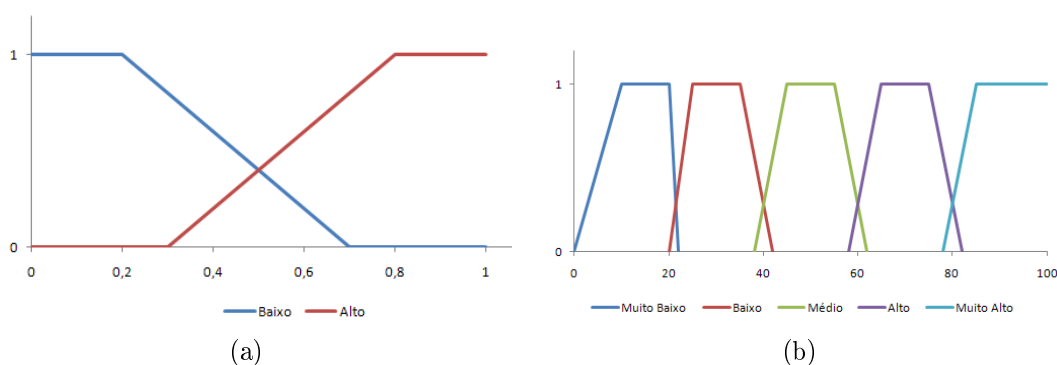


Figura 3.7: (a) Conjunto *fuzzy* do quantificador *most* e (b) conjunto *fuzzy* de risco de fogo

Deste modo, o processo de inferência de risco de fogo utiliza os dados locais fuzzificados e o quantificador *most* (opinião consensual sobre os dados da vizinhança) junto as regras *fuzzy* do apêndice B. Por exemplo, a regra 59 diz: "Se a temperatura local está alta e a umidade relativa está baixa e o grau de semelhança da temperatura da vizinhança é alto e o grau de semelhança da umidade relativa é alto então o risco

de fogo é alto". Assim como no autogerenciamento, o operador "e" e o sistema Mamdani são utilizados na ativação das regras e a operação *max*, na combinação das contribuições das regras ativadas. O método de defuzzificação Centróide é usado para detectar o grau de risco de fogo (Figura 3.7(b)).

A ontologia *Micro Sensornet* (μ SONG) de Hauifeng e Xingshe (2005) define as mensagens de contexto de risco de fogo enviadas ao sistema de monitoração do ambiente a cada intervalo de disseminação. A μ SONG expressa contexto em RSSF com apenas 3 atributos principais:

- ▶ *ContextName* é nome único de contexto. Seu valor neste trabalho é RISCO_FOGO;
- ▶ *ContextValue* é usado junto a *ContextName*. Seu valor neste trabalho é MUITO_ALTO;
- ▶ *Fidelity* indica a fidelidade ao *ContextValue*. Seu valor neste trabalho é o resultado da desfuzzificação, que de acordo com a Figura 3.7(b), isso representa um valor entre 0 e 100.

3.4 Conclusão

A MIAD presente em cada nó sensor de uma dada RSSF controla os intervalos de sensoriamento e disseminação bem como a redundância de informações que chegam ao sistema de monitoração do ambiente. A MIAD pretende diminuir a sobrecarga causada pela correlação de diferentes tipos de dados sensorizados e o grande número de mensagens que chegam ao sistema de monitoração. Ela realiza a função de agregação de *raw data* para composição de informações de contexto. Foi visto que o estudo de caso do risco de fogo, ilustra esse procedimento de composição, visto que o grau do risco de fogo obtido tem maior nível semântico do que os valores obtidos diretamente dos sensores (*raw data*) de temperatura e umidade relativa.

Ambos componentes da MIAD, gerente autônomo (MIA) e elemento gerenciado utilizam a lógica *fuzzy* por esta ser uma boa técnica de inferência robusta a imprecisão de informações vindas de sensores que podem estar mal calibrados ou defeituosos e também por ser computacionalmente leve, podendo ser usada por nós sensores. Encontram-se nos apêndices A e B as regras *fuzzy* dos procedimentos de inferência do autogerenciamento e da agregação de contexto de risco de fogo.

Espera-se, também, com o emprego da MIAD, a maximização da energia da RSSF, pois é previsível a diminuição da quantidade de mensagens enviadas por cada nó sensor ao sistema de monitoração do ambiente.

Resultados

Este capítulo descreve a metodologia utilizada para avaliar o desempenho da MIAD comparando com outras máquinas de inferência, assim, como os resultados obtidos nos experimentos. A Seção 4.1 apresenta uma descrição das máquinas de inferência implementadas e testadas. A Seção 4.2 mostra o cenário da RSSF testes onde as máquinas de inferência executaram. A Seção 4.3 são analisados e comparados parâmetros obtidos nos testes realizados que medem a funcionalidade das máquinas de inferência. Por fim, na Seção 4.4, são feitas algumas considerações sobre o capítulo.

4.1 Descrição das Máquinas de Inferência

Para validar a proposta, foram desenvolvidas 4 máquinas de inferência em linguagem nesC para o sistema operacional TinyOS:

- i. ESA - Elemento Sensor Autônomico que foi simulado no NS-2 de Braga (2006). Possui motor autônomico que usa regras rígidas para fazer a autoconfiguração do nó e o elemento gerenciado consiste em uma aplicação de sensoriamento de temperatura e umidade relativa do ambiente a qual envia mensagens de dados sensorizados localmente para o sistema de monitoração da RSSF;
- ii. MIAL - Máquina de Inferência Autônomico Localizada, cujo motor autônomico usa regras *fuzzy* para autoconfiguração do nó (ver apêndice A), conforme a descrição da Seção 3.2, porém, com o mesmo o elemento gerenciado do ESA, ou seja, uma aplicação de sensoriamento local;

- iii. DFLER - de Marin-Perianu e Havinga (2007) consiste em uma aplicação de sensoriamento distribuída que utiliza lógica *fuzzy* para inferenciar o grau de risco de fogo relacionando dados de temperatura e umidade relativa locais e da vizinhança. É semelhante ao elemento gerenciado descrito na Seção 3.3 e utiliza todas as regras *fuzzy* do apêndice B, mas, com intervalos de sensoriamento e disseminação fixos.
- iv. MIAD - Máquina de Inferência Autônoma Distribuída, proposta deste trabalho e descrita no Capítulo 3.

A Tabela 4.1 mostra os tipos de mensagens enviadas pelos 4 máquinas de inferência implementadas. Os intervalos de sensoriamento e disseminação de dados iniciam respectivamente em 20 e 40 segundos para o ESA, MIAL e MIAD. Para o DFLER esses intervalos são fixados em 40 segundos. No ESA, na MIAL e na MIAD, o ciclo autônomo se repete a cada 100 segundos e as mensagens de gerência são enviadas cada 200 segundos (intervalo de gerência).

Tabela 4.1: Tipos de Mensagens

Tipo	Tamanho (bytes)	Máquina de Inferência
Dados	10	ESA, MIAL
Gerência	22	ESA, MIAL, MIAD
<i>Fuzzy</i>	42	DFLER, MIAD
Contexto	8	DFLER, MIAD

4.2 Descrição do Cenário de Testes

A RSSF é composta por 6 nós sensores do tipo MICAz com placas de sensoriamento MTS400 (CROSSBOW, 2010) e um MICAz com uma aplicação *sniffer*. Este último é conectado a uma MIB520 para direcionar o tráfego da rede para um computador com a aplicação *SerialForwarder* em Java, que encaminha as mensagens da RSSF para uma outra aplicação em Java que as armazena em banco de dados para posterior análise.

Nenhum protocolo de roteamento foi utilizado. Mensagens de saúde (*health*) foram enviadas por todas as máquinas de inferência a cada minuto com informações

sobre o número de mensagens enviadas, mensagens perdidas e energia residual do nó sensor.

Cada máquina das 4 máquinas de inferência foi instalada nos 6 nós sensores para executar durante 1 hora. Com o propósito de analisar a reação das máquinas de inferência durante um possível risco de fogo, a RSSF ficou sob a ação de um secador de cabelo de 1700W durante 10 minutos. Neste cenário de teste, o evento de ar quente começa após os primeiros 100 segundos da ativação dos nós sensores. O evento de ar quente tanto modifica a temperatura como a umidade relativa da sala de testes resfriada a uma temperatura de 18 °C. O cenário de testes pode ser visto na Figura 4.1.



Figura 4.1: Testbed

Este procedimento foi repetido 10 vezes para cada máquina de inferência totalizando 40 horas de testes com o intuito de se analisar as 4 máquinas descritas através da média dos seguintes parâmetros:

- i. Consumo de memórias ROM;
- ii. Consumo de memória RAM;
- iii. Percentual de energia consumida pela RSSF;
- iv. Número total de mensagens enviadas;
- v. Número total de mensagens perdidas;
- vi. Número total de mensagens de dados;
- vii. Número total de mensagens de dados relevantes;

- viii. Número total de mensagens de contexto;
- ix. Número total de mensagens de contexto relevante;
- x. Quantidade de bytes enviados relacionados às mensagens de dados, dados relevantes, contexto e contexto relevante;
- xi. Valor dos intervalos médios e finais de sensoriamento e disseminação;
- xii. Valor da covariância destes intervalos;

As mensagens *health* e partes de código das máquinas de inferência que enviam tais mensagens não foram incluídos no cálculo dos parâmetros dos itens i e ii e nem dos itens iv até o item x supracitados. Isso porque as mensagens *health* auxiliam apenas no processo de avaliação individual do desempenho das máquinas de inferência implementadas.

Neste trabalho, um dado de temperatura é considerada relevante quando é maior ou igual a 30 °C e o dado de umidade relativa, quando é menor ou igual a 40% que indicam um grande aquecimento do ambiente (FUNCEME, 2010). O contexto de risco de fogo muito alto é relevante quando a fidelidade a esse contexto é maior ou igual a 70 (ver Seção 3.3).

4.3 Análise dos resultados

A Tabela 4.2 mostra a média dos nove primeiros parâmetros analisados e os outros parâmetros foram colocados em gráficos das figuras desta seção. Começando pelo consumo da memória, de todos as máquinas de inferência, a MIAD ocupa mais memória do sensor: cerca de 69,77% da ROM e 49,36% da RAM. É importante ressaltar que nenhum método perdeu mensagens e que o consumo de energia é praticamente o mesmo para MIAL e MIAD, que possuem a mesma máquina de inferência autônoma (MIA).

Comparando o consumo de energia das aplicações que enviam mensagens de dados ao sistema de monitoração da RSSF, foi visto que a MIAL economiza aproximadamente 16,23% da energia da RSSF a mais que o ESA. Foi observado que a MIAL envia um número total de mensagens 44,9% maior que o ESA, no entanto, ela envia 67,77% mais mensagens de dados relevantes que o ESA. Segundo a Figura 4.2, a média dos intervalos de sensoriamento e disseminação do ESA, como

também a média dos seus valores finais são um pouco maiores que a metade dos valores iniciais de 20 e 40 segundos, respectivamente.

Tabela 4.2: Média dos Parâmetros Analisados

Parâmetros	ESA	MIAL	DFLER	MIAD
Consumo de memória ROM (bytes)	23870	55666	54140	91452
Consumo de memória RAM (bytes)	1395	1622	1268	2022
Consumo de Energia (%)	1,17	0,98	1,20	0,98
Mensagens enviadas	499,7	907	2117,4	2009,2
Mensagens perdidas	0	0	0	0
Mensagens de Dados	99,6	504,8	-	-
Mensagens de Dados Relevantes	46,7	144,9	-	-
Mensagens de contexto	-	-	450,4	332,9
Mensagens de contexto relevantes	-	-	101,9	124,2

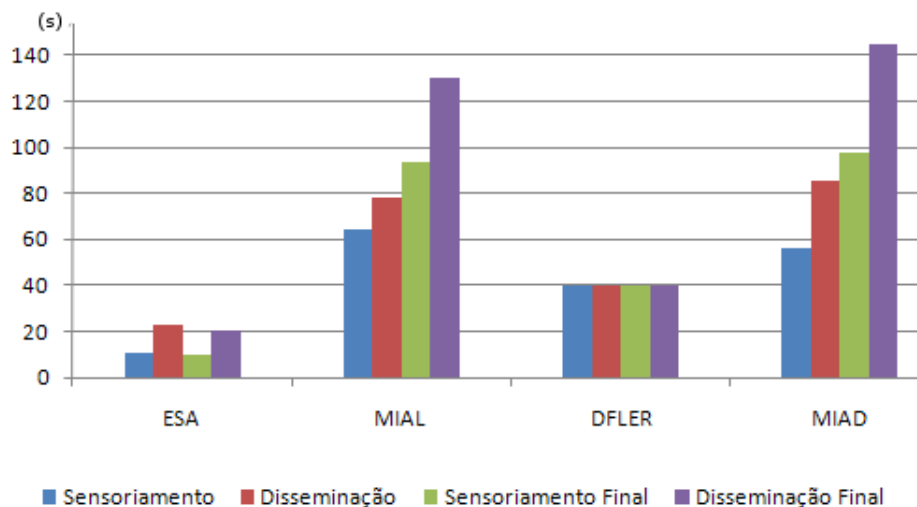


Figura 4.2: Média dos intervalos de sensoriamento e disseminação

A Figura 4.3 mostra a chegada de mensagens de dados do ESA ao sistema de monitoração em um teste das 13h22 às 14h22. Pode-se notar que o ESA soube diminuir o intervalo de disseminação, aumentado assim, a taxa de envio de mensagens de dados no momento do evento de ar quente, no entanto, fez com que todos os nós sensores parassem a disseminação antes mesmo do evento ser finalizado. Ou seja, houve inanição de mensagens de dados ao sistema de supervisão do ambiente e esse perfil de tráfego apresentado é bastante semelhante aos outros 9

testes feitos com o ESA. Os componentes de sensoriamento do TinyOS continuaram a ser chamados em pequenos intervalos de, em média, 10 segundos até o fim do teste, utilizando assim a maior parte da energia consumida para o sensoriamento.

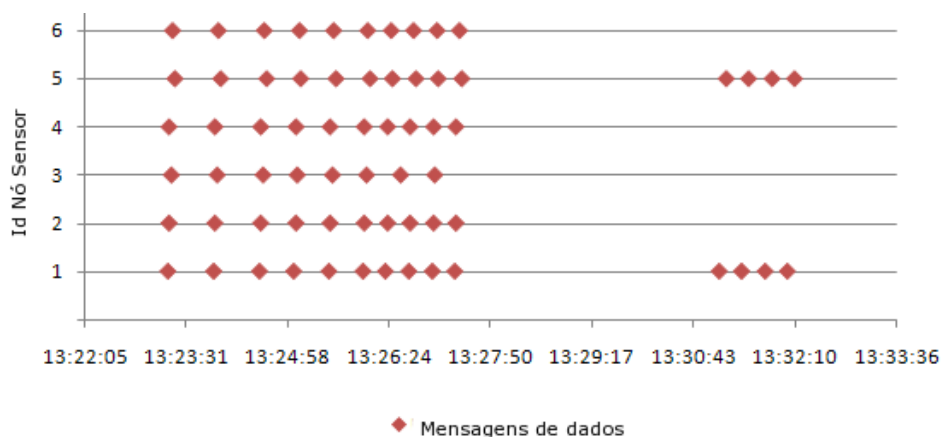


Figura 4.3: Mensagens de dados enviadas por cada nó sensor com ESA entre 13h22 e 14h22

A Tabela 4.2 ainda mostra que no caso das aplicações que enviam mensagens de contexto, a MIAD gastou 18,43% menos energia que o DFLER. A MIAD enviou 5,11% menos mensagens no total, 26,08% menos mensagens de contexto e um número de mensagens de contexto relevante 21,88% maior que o DFLER. Isso se deve ao fato da MIAD diminuir os intervalos de sensoriamento e disseminação no momento do evento de ar quente e de aumentá-los após o desaparecimento do evento. Na Figura 4.4, pode-se ver a chegada das mensagens de contexto da MIAD ao sistema de supervisão do ambiente em um teste realizado das 15h09 às 16h09. Nota-se a grande intensidade do fluxo de mensagens no início do teste correspondendo a reação da MIAD ao evento. Percebe-se ainda que o nó sensor 6 parou a disseminação de contexto por quase 20 minutos, mas logo depois voltou a disseminar a uma taxa menor devido ao término do evento e ao resfriamento do ambiente.

A Figura 4.5 mostra a autoconfiguração dos intervalos através da média do coeficiente de variância desses intervalos sendo os da MIAD e da MIAL maiores comparando com os do ESA. Os intervalos médios de sensoriamento e disseminação da MIAD são respectivamente 5,67 e 3,68 vezes maiores do que os do ESA e seus valores finais médios são respectivamente, 9,72 e 7,12 maiores, segundo a Figura 4.2. Os resultados demonstram que a MIAD e a MIAL aumentam as taxas de sensoriamento e de disseminação durante o evento de ar quente e diminui essas taxas quando o evento desaparece, economizando desta maneira a energia da RSSF

e diminuindo a quantidade de contextos irrelevantes do risco de fogo muito alto, sem causar inanição de mensagens de informações enviadas ao sistema de monitoração da RSSF.

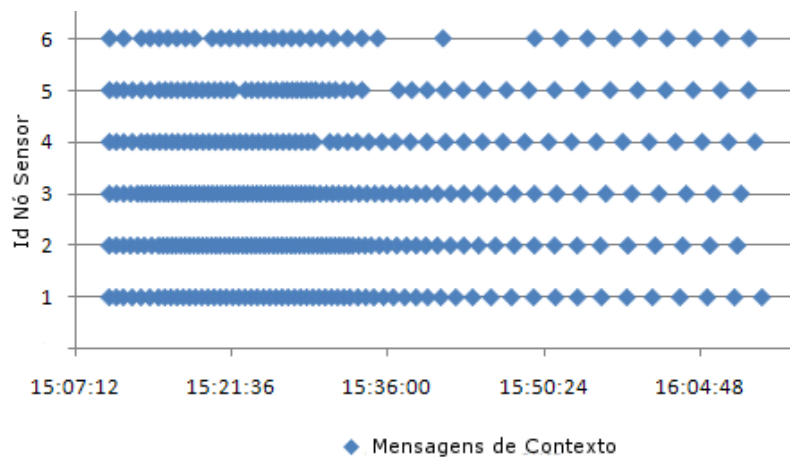


Figura 4.4: Mensagens de contexto enviadas por cada nó sensor com a MIAD entre 15h09 e 16h09

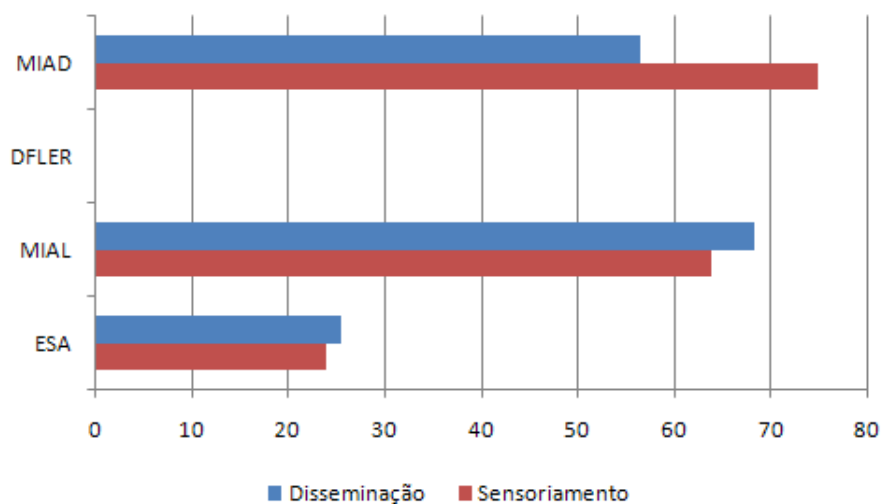


Figura 4.5: Média do coeficiente de variância dos intervalos de disseminação e sensoriamento

Conforme a Figura 4.6, os resultados também mostram que a MIAD envia ao sistema de monitoração do ambiente 47,24% menos bytes de mensagens destinadas ao sistema de monitoração da RSSF do que a MIAL e também 31,42% menos bytes de informações relevantes que a MIAL. Isto se deve ao fato da mensagem de contexto ser menor que a mensagem de dados, além de possuir nível semântico maior. As mensagens de contexto baseadas na ontologia μ SONG representam o conhecimento da RSSF sobre as condições atuais do ambiente monitorado, o que diminui de certa

forma a sobrecarga que o sistema de monitoração da RSSF teria para agregar todos os dados de temperatura e umidade relativa em informações de contexto de risco de fogo, pois, isto, a RSSF faz de maneira distribuída.

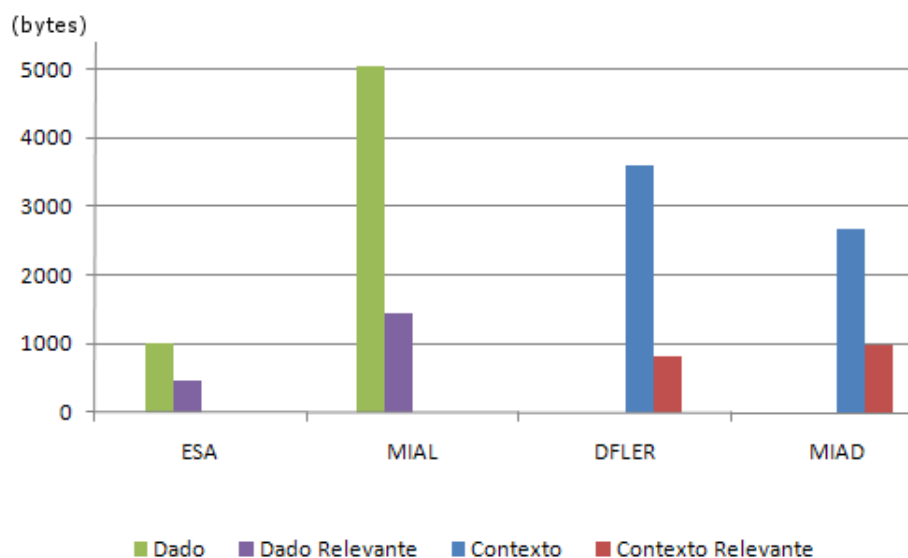


Figura 4.6: Média de bytes das mensagens enviadas ao sistema de monitoração do ambiente

4.4 Conclusão

Os experimentos descritos neste capítulo foram feitos com recursos disponíveis, tais, como nós sensores reais da empresa (CROSSBOW, 2010), o secador de cabelo de 1700 W, um notebook e sala com ar-condicionado. A ação do evento de ar quente não foram maiores em virtude do aquecimento do secador de cabelo que o fazia parar. Era necessário pelo menos 1h de pausa no funcionamento do secador antes de cada reinício dos experimentos com os sensores.

Não houve simulações do cenário utilizando o TOSSIM ou NS-2 que permitiriam a medição da escalabilidade da proposta. No entanto, apesar dos testes terem sido executados em um cenário real simplista, notou-se a importância da MIAD para RSSF de monitoração de ambiente, pois, considerou-se significativa a redução de mensagens enviadas ao sistema de monitoração e a economia da energia da RSSF.

Considerações Finais

Esta dissertação propôs a MIAD, uma máquina de inferência autônoma distribuída para RSSF. Na Seção 5.1, são apresentadas conclusões obtidas a partir dos resultados do capítulo anterior. A Seção 5.2 relaciona as principais contribuições deste trabalho e a Seção 5.3 lista vários trabalhos futuros que podem ser realizados a partir desta dissertação.

5.1 Conclusões

A MIAD pode contribuir na construção de sistemas de monitoração orientados a Inteligência de Ambiente, pois provê informações de contexto com maior valor semântico do que os *raw data*. Aplicações de sensoriamento são muito dependentes dos sensores. Portanto, foi apresentado nesta dissertação um estudo de caso que leva em consideração a agregação ou composição semântica de *raw data* de nós sensores disponíveis de temperatura e umidade relativa em informações de contexto de risco de fogo.

Ao mesmo tempo, a MIAD autogerencia os intervalos de sensoriamento e disseminação, diminuindo-os em momentos de risco de fogo muito alto, representados nos testes por eventos de ar quente, e aumentando-os nos momentos em que o risco de fogo diminui (ausência do evento). A minimização da redundância de informações da RSSF é feita parando a disseminação de mensagens ao sistema de monitoração do ambiente, quando a MIAD detecta que as informações sensoriadas representam ausência de risco de fogo muito alto e as configurações do nó sensor são semelhantes a de seus vizinhos.

O uso da Inteligência Artificial é incentivado em tecnologias da Inteligência de Ambiente, como as RSSF e nas funções de autogerenciamento da Computação Autônoma. Portanto, a lógica *fuzzy* é de grande importância e atua em ambos serviços da MIAD: autoconfiguração e produção distribuída de contexto feita pelos nós sensores. Essa técnica de inferência é computacionalmente leve, fácil de se implementar nos nós sensores e também representa um mecanismo robusto para lidar informações sensorizadas que podem vir de sensores defeituosos ou imprecisos.

Os resultados mostraram que a MIAD envia menos mensagens de informações de contexto ao sistema de monitoração do ambiente do que o DFLER, que não possui autogerenciamento. Porém, durante os eventos de ar quente, foi a máquina que mais enviou mensagens ao sistema caracterizando melhor a situação de risco de fogo. Por isso, a MIAD economizou mais energia que o DFLER.

As mensagens de contexto são menores que as de dados enviadas pelas outras máquinas de inferência comparadas (MIAL e ESA), porém com maior valor semântico. Apesar da MIAD possuir o mesmo gerente autônomo que a MIAL, ela tem, como elemento gerenciado, uma aplicação distribuída produtora de contexto, semelhante ao DFLER. Por isso, deduz-se que MIAD e MIAL consumiram quase a mesma quantidade de energia, uma vez que, a MIAD gastou menos energia para enviar as mensagens de contexto (menores que as de dados da MIAL) ao sistema de monitoração do ambiente, porém, gastou um pouco mais de energia na produção distribuída de contexto.

Considera-se limitações desta proposta o cenário de testes, pois, apesar de terem sido utilizados, nós sensores reais, o número de motes MICAz foi baixo (apenas 6 nós). Além disso, estavam muito próximos um dos outros e em uma sala fechada sem ruídos, não representando muito bem um cenário real, como o de uma cidade, ou reserva florestal, ilustrados pelos eSENSE (2007) e SENSEI (2010).

5.2 Contribuições

A principal contribuição desta dissertação foi o desenvolvimento da máquina de inferência autônoma distribuída (MIAD) para RSSF. A MIAD usa lógica *fuzzy* para produção de informações de contexto e funções de autogerenciamento. Ela produz informações de contexto de nível semântico mais alto que os dados vindos diretamente dos sensores (*raw data*) de uma RSSF. Em paralelo a produção de contexto, a MIAD promove o autogerenciamento do nó sensor maximizando a

vida da RSSF através de intervalos de sensoriamento e disseminação inteligentes, além da autoconfiguração da redundância de informações de contexto, diminuindo a quantidade de mensagens que chegam ao sistema de monitoração do ambiente. A MIAD foi avaliada em um cenário com nós sensores da empresa (CROSSBOW, 2010) e comparada a outras 3 máquinas de inferência, sendo 2 destas preocupadas com o autogerenciamento do nó sensor.

Os códigos em linguagem nesC da implementação das 4 máquinas de inferência (ESA, MIAL, DFLEER e MIAD) são disponibilizadas a pesquisadores interessados. Os códigos funcionam em motes MICAz e placa de sensoriamento MTS400 com TinyOS 1.X. O ESA foi desenvolvido com base na dissertação de Braga (2006), de acordo com instância simulada para o NS-2. Foi feita apenas uma adaptação para que o ESA tratasse dois tipos de informações sensoriadas (temperatura e umidade relativa), pois na instância do NS-2, foi utilizada apenas um tipo de informação (monóxido de carbono - CO).

Marin-Perianu e Havinga (2007) utilizaram informações de temperatura e nível de fumaça para que os nós sensores com o DFLEER produzissem contexto de "decisão de fogo" para sensores μ Nodes da empresa Ambient (2010). Na versão para sensores da Crossbow (2010), foi substituída a informação de fumaça pela umidade relativa e a função *most* utilizada é definida pelo conjunto *fuzzy* da Figura 3.7(a). Na ausência de mensagens *fuzzy* dos vizinhos, o nó sensor é capaz de inferenciar o contexto apenas com os dados sensoriados localmente, detalhe este que não foi mencionado em (MARIN-PERIANU; HAVINGA, 2007), no entanto, foi implementado na versão em nesC.

Neste trabalho, ambas implementações da MIAD e do DFLEER enviam mensagens de contexto baseadas na ontologia μ SONG (HAUIFENG; XINGSHE, 2005), a qual expressa o conhecimento de risco de fogo da RSSF. Todas regras *fuzzy* de autoconfiguração do apêndice A foram implementadas pela MIAL e MIAD, enquanto que as regras *fuzzy* de decisão de risco de fogo do apêndice B foram implementadas pela MIAD e DFLEER.

Vale ressaltar que as variáveis linguísticas e regras *fuzzy* de autoconfiguração mais o gerente autônomo, MIA, da Seção 3.2, representam uma contribuição genérica desse trabalho. E as variáveis linguísticas e regras *fuzzy* usados pelo elemento gerenciado da Seção 3.3 representam uma contribuição dependente de sensores de temperatura e umidade relativa.

Os resultados parciais foram aceitos em evento internacional através do seguinte artigo: "Distributed Autonomic Inference Machine for Wireless Sensor Networks", Nídia Campos, Danielo Gomes, José de Souza, Proceedings of International Telecommunications Symposium (ITS) 2010. SEPTEMBER 06-09, 2010, Manaus, Amazonas, Brazil. <<http://telematica.ee.ufpe.br/its2010/>>.

5.3 Trabalhos Futuros

É importante testar a MIAD na produção de outros de tipos de contexto. Para isso, conhecimento especialista, como o do meteorologista da FUNCEME (2010), será necessário para modelagem de regras *fuzzy* que serão implementadas em cada nó sensor que monitora o ambiente em questão.

Nos sistemas *fuzzy* associados ao gerente autônomo e elemento gerenciado da MIAD, foi utilizado apenas o método centro da área (Centróide) no procedimento de desfuzzificação. Existem outros tipos de métodos de desfuzzificação como o critério do máximo, média do máximo e média ponderada dos máximos que devem ser explorados com a finalidade de se obter melhores resultados no processo inferência de informações de contexto e no autogerenciamento do nó sensor (ADELI; SARMA, 2006). As mesmas variáveis linguísticas podem ser utilizadas nesse processo, como também pode-se modelar novamente essas variáveis usando outros tipos de funções. Por exemplo, usar funções triangulares, no lugar de funções trapezoidais e vice-versa e se verificar o desempenho depois dessas mudanças.

Algoritmos de clusterização podem ser adicionados a MIAD com o objetivo de aumentar a vida e escalabilidade de RSSF. Por exemplo, Rocha et al. (2010) propõem um modelo de clusterização semântica para encontrar o relacionamento da vizinhança em RSSF de monitoração da saúde de estruturas.

A MIAD foi executada em uma RSSF com apenas 6 nós sensores da (CROSSBOW, 2010) em um cenário sem ruídos (sala fechada) como descrito no Capítulo 4. Logo, é importante simulá-la em uma RSSF com dezenas de nós sensores para verificar melhor o parâmetro de perda de pacotes e energia consumida, adicionando ruídos aos cenários simulados. O TOSSIM é uma boa opção, pois, simula aplicações do sistema operacional TinyOS escritas em linguagem nesC (TINYOS, 2010b). Seus cenários são escritos em linguagem Python, mas, existe uma GUI chamada JTOSSIM, que facilita a configuração de parâmetros da simulação e visualização dos resultados (KELLER, 2010).

Com um número maior de nós sensores, torna-se necessário o emprego de um protocolo de roteamento para RSSF para que cada nó sensor contribua com o transporte das mensagens de contexto até o sistema de monitoração do ambiente. A Crossbow (2007c) oferece o protocolo XMesh que escolhe o melhor caminho através de uma função custo que leva em consideração a qualidade de envio e recepção dos enlaces e também do custo do nó sensor vizinho.

A MIAD também pode ser usada em conjunto com o PROST de Neto, Neto e Andrade (2010), um protocolo de roteamento sensível ao tempo, que permitiria a entrega do contexto de risco de fogo produzido na RSSF a sistemas de monitoração de ambientes críticos e perigosos. Outra opção interessante de protocolo de roteamento é o BIO4SeL de Ribeiro e Castro (2010). Ele é baseado em colônias de formigas, uma técnica de Inteligência de Enxame (*Swarm Intelligence*) e, assim como a MIAD, usa princípios da Computação Autônômica para otimizar a energia da RSSF.

A migração do código para o TinyOS 2.X é importante, visto que algoritmos de roteamento para RSSF que antes só existiam em simuladores, como o NS-2, estão sendo implementados nesta nova versão do TinyOS, por exemplo, o protocolo de roteamento AODV (KIM, 2008).

A MIAD envia informações de contexto ao sistema de monitoração do ambiente de forma assíncrona, ou seja, de acordo com o intervalo de disseminação. Uma outra abordagem que pode ser implementada junto a MIAD é o processamento de consultas (*queries processing*) de contexto em RSSF. Nessa abordagem síncrona de envio de informações, são levadas em consideração as restrições de memória e energia de RSSF (BRAYNER et al., 2008).

Outra sugestão de trabalho futuro é a integração da RSSF com a MIAD a aplicações móveis sensíveis ao contexto através de *middlewares*, como ConBus (*Context Bus*), que provê, para a plataforma Android, contexto vindos de sensores em geral, incluindo aqueles de nós sensores de uma RSSF (Sá et al., 2010).

Base de Conhecimento de Autoconfiguração

A Tabela A.1 relaciona as seguintes variáveis linguísticas para autoconfiguração dos intervalos de sensoriamento e disseminação:

- ▶ Percentagem de Dados Relevantes - perDadosRel {**Baixa**, **Média**, **Alta**} da Figura 3.2(a);
- ▶ Número de Bytes Transmitidos - numBytesTrans {**Baixo**, **Médio**, **Alto**} da Figura 3.2(b);
- ▶ Percentagem de Energia Residual - perEnergResidual {**Baixa**, **Média**, **Alta**} da Figura 3.2(a);
- ▶ Intervalo de Sensoriamento Atual - interSenso {**Baixo**, **Médio**, **Alto**} da Figura 3.3(a);
- ▶ Configuração do Intervalo de Sensoriamento - configInterSenso {**Muito Alto**, **Alto**, **Adequado**, **Razoavelmente Baixo**, **Baixo**} da Figura 3.5(a);
- ▶ Configuração do Intervalo de Disseminação - configInterDisse {**Alto**, **Razoavelmente Alto**, **Adequado**, **Baixo**} da Figura 3.5(b).

A Tabela A.2 relaciona as mesmas variáveis linguísticas para autoconfiguração dos intervalos de sensoriamento e disseminação que a Tabela A.1 com acréscimo da variável Intervalo de Disseminação Atual - interDisse {**Baixo**, **Médio**, **Alto**} da

Figura 3.3(b).

Tabela A.1: Regras para Autoconfiguração de Intervalo de Sensoriamento e de Disseminação (Parte 1)

Regra	perDadosRel	numBytesTrans	perEnergResidual	interSenso	configInterSenso	configInterDisse
1	A	A	A	A	MA	AD
2	A	A	M	A	MA	AD
3	M	A	A	A	MA	AD
4	M	A	M	A	MA	AD
5	A	A	A	M	MA	AD
6	A	A	M	M	MA	AD
7	M	A	A	M	MA	AD
8	M	A	M	M	MA	AD

Tabela A.2: Regras para Autoconfiguração de Intervalo de Sensoriamento e de Disseminação (Parte 2)

Regra	perDadosRel	numBytesTrans	perEnergResidual	interSenso	interDisse	configInterSenso	configInterDisse
9	A	M	A	A	A	A	A
10	A	M	A	A	M	A	A
11	A	M	A	M	A	A	A
12	A	M	A	M	M	A	A
13	A	M	M	A	A	A	A
14	A	M	M	A	M	A	A
15	A	M	M	M	A	A	A
16	A	M	M	M	M	A	A
17	A	B	A	A	A	A	A
18	A	B	A	A	M	A	A
19	A	B	A	M	A	A	A
20	A	B	A	M	M	A	A
21	A	B	M	A	A	A	A
22	A	B	M	A	M	A	A
23	A	B	M	M	A	A	A
24	A	B	M	M	M	A	A
25	M	M	A	A	A	A	A
26	M	M	A	A	M	A	A
27	M	M	A	M	A	A	A
28	M	M	A	M	M	A	A
29	M	M	M	A	A	A	A
30	M	M	M	A	M	A	A
31	M	M	M	M	A	A	A
32	M	M	M	M	M	A	A
33	M	B	A	A	A	A	A
34	M	B	A	A	M	A	A
35	M	B	A	M	A	A	A
36	M	B	A	M	M	A	A
37	M	B	M	A	A	A	A
38	M	B	M	A	M	A	A
39	M	B	M	M	A	A	A
40	M	B	M	M	M	A	A

A Tabela A.3 relaciona as seguintes variáveis linguísticas para autoconfiguração dos intervalos de sensoriamento e de disseminação:

- ▶ Percentagem de Dados Relevantes dos Vizinhos - perDadosRelViz {**B**aixa, **M**édia, **A**lta} da Figura 3.2(a);
- ▶ perDadosRel, interSenso, InterDisse, configInterSenso e configInterDisse já descritas neste apêndice.

Tabela A.3: Regras para Autoconfiguração de Intervalo de Sensoriamento e de Disseminação (Parte 3)

Regra	perDadosRelViz	perDadosRel	interSenso	interDisse	configInterSenso	configInterDisse
41	B	B	A	M	B	B
42	B	B	A	B	B	B
43	B	B	M	M	B	B
44	B	B	M	B	B	B
45	A	B	A	A	AD	RA
46	A	B	M	A	AD	RA
47	M	B	A	A	AD	RA
48	M	B	M	A	AD	RA
49	A	B	A	M	AD	RA
50	A	B	M	M	AD	RA
51	M	B	A	M	AD	RA
52	M	B	M	M	AD	RA
54	B	B	B	M	RB	B
55	B	B	B	B	RB	B
57	A	B	B	A	RB	RA
58	M	B	B	A	RB	RA
59	A	B	B	M	RB	RA
60	M	B	B	M	RB	RA
61	A	B	B	B	B	B
62	A	M	B	B	B	B
63	A	B	M	B	AD	B
64	A	M	M	B	AD	B
65	M	M	M	B	AD	B

A Tabela A.4 relaciona as seguintes variáveis linguísticas para autoconfiguração de redundância de informações:

- ▶ Percentagem de Nós Vizinhos de Configuração Semelhante - perViziConfigSem {Baixa, Média, Alta} da Figura 3.4;
- ▶ perDadosRel, já descrita neste apêndice;
- ▶ Redundância de Informações - redundância {Baixa, Média, Alta} da Figura 3.4.

Tabela A.4: Regras para Autoconfiguração da Redundância de Informações

Regra	perViziConfigSem	perDadosRel	redundância
66	A	A	B
67	A	M	M
68	A	B	A
69	M	A	B
70	M	M	M
71	M	B	M
72	B	A	B
73	B	M	B
74	B	B	A

Apêndice **B**

Base de Conhecimento de Risco de Fogo

Um metereologista da FUNCEME (2010) foi entrevistado sobre como detectar risco de fogo/incêndio utilizando dados de temperatura e umidade relativa. Ele respondeu o seguinte questionário:

Questão 1. Indique um intervalo no qual a temperatura pode ser classificada como:

- i. Muito Baixa (MB). *Resposta: valores $< 21,0^{\circ}C$.*
- ii. Baixa (B). *Resposta: valores de $21,1$ a $23,9^{\circ}C$.*
- iii. Média(M). *Resposta: valores de $24,0$ a $25,9^{\circ}C$.*
- iv. Alta (A). *Resposta: valores de $26,0$ a $29,9^{\circ}C$.*
- v. Muito Alta (MA). *Resposta: valores de $> 30,0^{\circ}C$.*

OBS: um trecho do intervalo pode ser classificado/pertencer até dois dos conjuntos acima.

Questão 2. Indique um intervalo no qual a umidade relativa pode ser classificada como:

- i. Muito Baixa (MB). *Resposta: valores $< 40,0\%$.*

- ii. Baixa (B). *Resposta: valores de 40,1 a 49,9 %.*
- iii. Média (M). *Resposta: valores de 50,0 a 59,9 %.*
- iv. Alta (A). *Resposta: valores de 60,0 a 79,9 % .*
- v. Muito Alta (MA). *Resposta: valores de 80,0 a 100,0 %.*

OBS: um trecho do intervalo pode ser classificado/pertencer até dois dos conjuntos acima.

Questão 3. De acordo com as questões 1 e 2, complete a tabela abaixo classificando o risco de fogo/incêndio com MB para muito baixo, B para baixo, M para Médio, A para alto e MA para muito Alta.

Tabela B.1: Conhecimento sobre Risco de Fogo

Temperatura	Umidade Relativa	Risco de Fogo
MB	MB	
MB	B	
MB	M	
MB	A	
MB	MA	
B	MB	
B	B	
B	M	
B	A	
B	MA	
M	MB	
M	B	
M	M	
M	A	

(continua)

Temperatura	Umidade Relativa	Risco de Fogo
M	MA	
A	MB	
A	B	
A	M	
A	A	
A	MA	
MA	MB	
MA	B	
MA	M	
MA	A	
MA	MA	

Como pode ser visto, apenas as duas primeiras questões foram respondidas pelo meteorologista, pois, segundo ele, a FUNCEME, não tem experiência na detecção de risco de fogo. Deste modo, para se completar o conhecimento sobre tal risco, foram realizados experimentos com os sensores de temperatura e umidade relativa da Crossbow (2010) sob ação de eventos de ar quente produzidos por um secador de cabelos de 1700 W em uma sala resfriada a 18 °C (ver Figura 4.1).

Assim, com as respostas das questões 1 e 2 e as observações dos experimentos, pode-se montar a base de conhecimento, que consiste na Tabela B.2 que relaciona as seguintes variáveis linguísticas usadas em regras para produção de contexto de risco de fogo:

- ▶ Temperatura - T {**Muito Baixa**, **Baixa**, **Média**, **Alta**, **Muito Alta**} da Figura 3.6(a);
- ▶ Umidade Relativa - UR {**Muito Baixa**, **Baixa**, **Média**, **Alta**, **Muito Alta**} da Figura 3.6(b) ;
- ▶ Grau de Semelhança da Temperatura dos Vizinhos - $most(T_{viz})$ {**Baixo**, **Alto**} da Figura 3.7(a);

- ▶ Grau de Semelhança da Umidade Relativa dos Vizinhos - $most(UR_{viz})$ {Baixo, Alto} da Figura 3.7(a);
- ▶ Grau de Risco e Fogo - RF da Figura 3.7(b).

Tabela B.2: Regras para Produção de Contexto de Risco de Fogo

Regra	T	UR	$most(T_{viz})$	$most(UR_{viz})$	RF
1	MB	MB	B	B	MB
2	MB	MB	B	A	MB
3	MB	MB	A	B	MB
4	MB	MB	A	A	MB
5	MB	B	B	B	MB
6	MB	B	B	A	MB
7	MB	B	A	B	MB
8	MB	B	A	A	MB
9	MB	A	B	B	MB
10	MB	A	B	A	MB
11	MB	A	A	B	MB
12	MB	A	A	A	MB
13	B	MB	B	B	B
14	B	MB	B	A	B
15	B	MB	A	B	B
16	B	MB	A	A	B
17	B	B	B	B	MB
18	B	B	B	A	B
19	B	B	A	B	B
20	B	B	A	A	B

(continua)

Regra	T	UR	$most(T_{viz})$	$most(UR_{viz})$	RF
21	B	M	B	B	MB
22	B	M	B	A	MB
23	B	M	A	B	MB
24	B	M	A	A	MB
25	B	A	B	B	MB
26	B	A	B	A	MB
27	B	A	A	B	MB
28	B	A	A	A	MB
29	B	MA	B	B	MB
30	B	MA	B	A	MB
31	B	MA	A	B	MB
32	B	MA	A	A	MB
33	M	MB	B	B	M
34	M	MB	B	A	A
35	M	MB	A	B	A
36	M	MB	A	A	A
37	M	B	B	B	B
38	M	B	B	A	M
39	M	B	A	B	M
40	M	B	A	A	M
41	M	M	B	B	MB
42	M	M	B	A	B
43	M	M	A	B	B
44	M	M	A	A	B
45	M	A	B	B	MB

(continua)

Regra	T	UR	$most(T_{viz})$	$most(UR_{viz})$	RF
46	M	A	B	A	B
47	M	A	A	B	B
48	M	A	A	A	B
49	M	MA	B	B	MB
50	M	MA	B	A	B
51	M	MA	A	B	B
52	M	MA	A	A	B
53	A	MB	B	B	A
54	A	MB	B	A	MA
55	A	MB	A	B	MA
56	A	MB	A	A	MA
57	A	B	B	B	B
58	A	B	B	A	A
59	A	B	A	B	A
60	A	B	A	A	A
61	A	M	B	B	B
62	A	M	B	A	M
63	A	M	A	B	M
64	A	M	A	A	M
65	A	A	B	B	B
66	A	A	B	A	M
67	A	A	A	B	M
68	A	A	A	A	M
69	A	MA	B	B	B
70	A	MA	B	A	M

(continua)

Regra	T	UR	$most(T_{viz})$	$most(UR_{viz})$	RF
71	A	MA	A	B	M
72	A	MA	A	A	M
73	MA	MB	B	B	M
74	MA	MB	B	A	MA
75	MA	MB	A	B	MA
76	MA	MB	A	A	MA
77	MA	B	B	B	M
78	MA	B	B	A	A
79	MA	B	A	B	A
80	MA	B	A	A	A
81	MA	M	B	B	B
82	MA	M	B	A	M
83	MA	M	A	B	M
84	MA	M	A	A	M
85	MA	A	B	B	MB
86	MA	A	B	A	MB
87	MA	A	A	B	MB
88	MA	A	A	A	MB
89	MA	MA	B	B	MB
90	MA	MA	B	A	MB
91	MA	MA	A	B	MB
92	MA	MA	A	A	MB

Referências Bibliográficas

ACHARJEE, D.; PATEL, P. Multipath routing sensor network for finding crack in metallic structure using fuzzy logic. In: *World Academy of Science, Engineering and Technology*. [S.l.: s.n.], 2008. v. 36, p. 324–331.

ADELI, H.; SARMA, K. C. *Cost Optimization of Structures: Fuzzy Logic, Genetic Algorithms, and Parallel Computing*. [S.l.]: Willey, 2006.

ALLIANCE, Sensor Web. *Sensor Web Concept*. 2010. Disponível em: <<http://sites.google.com/site/sensorweballiance/Home/sensor-web-concept>>.

AMBIENT. *Environmental Monitoring through Wireless Sensor Networks*. 2010. Disponível em: <<http://ambient-systems.net>>.

AUGUSTO, J. C. Ambient intelligence: the confluence of ubiquitous/pervasive computing and artificial intelligence. In: . [S.l.]: Springer London, 2007. p. 213–234.

AZIM, M. A.; JAMALIPOUR, A. Optimized forwarding for wireless sensor networks by fuzzy inference system. In: *International Conference on Wireless Broadband and Ultraband*. [S.l.: s.n.], 2006.

BALDAUF, M.; DUSTDAR, S.; ROSENBERG, F. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2004.

BRAGA, T. R. *Um elemento autônomo para redes de sensores sem fio*. Dissertação (Mestrado) — Universidade Federal de Minas Gerais, 2006. Disponível em: <<http://www.bibliotecadigital.ufmg.br/dspace/handle/1843/SLBS-6Y5JFK>>.

BRAYNER, Angelo et al. An adaptive in-network aggregation operator for query processing in wireless sensor networks. *J. Syst. Softw.*, Elsevier Science Inc., New York, NY, USA, v. 81, n. 3, p. 328–342, 2008. ISSN 0164-1212.

COPETTI, A.; LEITE, J. C. B. Intelligent context-aware monitoring in home care. *IEEE 2nd International Conference: E-Medical Systems*, 2008.

CROSSBOW. Mpr - mib users manual. 2007.

CROSSBOW. Mts/mda sensor board users manual. 2007.

CROSSBOW. *XMesh User's Manual*. 2007.

CROSSBOW. Wireless measurement system. 2010. Disponível em: <<http://www.xbow.com>>.

CRUZ, Shermila Guerra Santa. *Sistema de Navegação para Veículo Autônomo utilizando Lógica Difusa*. Dissertação (Mestrado) — Escola de Engenharia de São Carlos - Universidade de São Paulo (USP), 2009.

DEY, Anind K. Understanding and using context. *Personal Ubiquitous Computing*, Springer-Verlag, London, UK, v. 5, n. 1, p. 4–7, 2001. ISSN 1617-4909.

ESENSE. Capturing ambient intelligence for mobile communications through wireless sensor networks. *European 6th framework Programme for Research and Technological Development*, 2007. Disponível em: <<http://www.ist-esense.org>>.

FUNCEME. *Fundação Cearense de Meteorologia e Recursos Hídricos*. 2010. Disponível em: <<http://www.funceme.br>>.

GOMES, R. L. et al. Maximização da vida útil de redes de sensores sem fio utilizando fusão de dados e roteamento fuzzy. In: 28º SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E DE SISTEMAS DISTRIBUÍDOS - SBRC. *XV Workshop de Gerência e Operação de Serviços*. [S.l.]: Sociedade Brasileira de Computação - SBC, 2010. p. 185–197.

GOMIDE, F. A. C.; GUDWIN, R. R. Modelagem, controle, sistemas e lógica fuzzy. *SBA Controle & Automação*, v. 4, n. 3, p. 97–115, setembro-outubro 1994. Disponível em: <<ftp://ftp.dca.fee.unicamp.br/pub/docs/gudwin/publications/RevSBA94.pdf>>.

HAUIFENG, Q.; XINGSHE, Z. Context aware sensor net. In: *3rd International Workshop on Middleware for pervasive and ad-hoc computing*. [S.l.: s.n.], 2005. (ACM International Conference Proceeding Series, v. 115), p. 1–7.

HORN, P. *Autonomic Computing: IBM's Perspective on the State of Information Technology*. 2001. Disponível em: <http://www.research.ibm.com/autonomic/manifesto/autonomic_computing.pdf>.

KAWASHIMA, H. et al. Met: a real world oriented metadata management system for semantic sensor networks. In: *DMSN '06: Proceedings of the 3rd workshop on Data management for sensor networks*. New York, NY, USA: ACM, 2006. p. 13–18.

KELLER, Lorenzo. *JTossim*. 2010. Disponível em: <<http://sourceforge.net/projects/jtossim>>.

KEPHART, J. O.; CHESS, D. M. The vision of autonomic computing. *Computer*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 36, n. 1, p. 41–50, 2003. ISSN 0018-9162.

KIM, J. *AODV implementation on TinyOS-2.x*. 2008. Disponível em: <<http://usn.konkuk.ac.kr/~jskim/AODV.html>>.

MARIN-PERIANU, M. *Collaborative Wireless Sensor Networks in Industrial and Business Processes*. Tese (Doutorado) — University of Twente, November 2008.

MARIN-PERIANU, M.; HAVINGA, P. J. M. D-fler - a distributed fuzzy logic engine for rule-based wireless sensor networks. In: ICHIKAWA, Haruhisa et al. (Ed.). *UCS*. [S.l.]: Springer, 2007. (Lecture Notes in Computer Science, v. 4836), p. 86–101. ISBN 978-3-540-76771-8.

MARIN-PERIANU, M. et al. Distributed activity recognition with fuzzy-enabled wireless sensor networks. In: *DCOSS '08: Proceedings of the 4th IEEE international conference on Distributed Computing in Sensor Systems*. Berlin, Heidelberg: Springer-Verlag, 2008. p. 296–313. ISBN 978-3-540-69169-3.

MARIN-PERIANU, M. et al. Be aware in wireless sensor networks. *15th IST Mobile & Wireless Communication Summit*, 2006.

MCCARTHY, J. D. *Using Sensor Ontologies to create Reasoning-Ready Sensor Data for Real-time Hazard Monitoring in Spatial Decision Support System*. Dissertação (Mestrado) — Universidade de Windsor, Ontario - Canadá, 2007.

MERICO, D.; MILEO, A.; BISIANI, R. Wireless sensor networks supporting context-aware reasoning in assisted living. In: *PETRA '08: Proceedings of the 1st international conference on Pervasive Technologies Related to Assistive Environments*. New York, NY, USA: ACM, 2008. p. 1–2. ISBN 978-1-60558-067-8.

MICHEL, Ricardo Cunha. *LAFIQ - Instrumentação, Sensores & Atuadores*. 2009. Disponível em: <<http://www.ima.ufrj.br/~rmichel/>>.

NETO, J. B. B.; NETO, P. F. R.; ANDRADE, R. M. C. A delay-sensitive strategy for real-time monitoring in wireless sensor networks. In: *IEEE 17th International Conference on Telecommunications - ICT*. [S.l.: s.n.], 2010. p. 814–821.

RAMOS, Carlos. Ambient intelligence - a state of the art from artificial intelligence perspective. In: *Progress in Artificial Intelligence*. [S.l.]: Springer Berlin / Heidelberg, 2007, (Lecture Notes in Computer Science, Volume 4874/2007). p. 285–295.

RAMOS, C.; AUGUSTO, J. C.; SHAPIRO, D. Ambient intelligence - the next step for artificial intelligence. *IEEE Intelligent Systems*, IEEE Computer Society, Los Alamitos, CA, USA, v. 23, p. 15–18, 2008. ISSN 1541-1672.

RIBEIRO, L. B.; CASTRO, M. F. de. Bio4sel: A bio-inspired routing algorithm for sensor network lifetime optimization. In: *IEEE 17th International Conference on Telecommunications - ICT*. [S.l.: s.n.], 2010. p. 728–734.

ROCHA, Atslands da et al. Semantic clustering in wireless sensor networks. In: PONT, Ana; PUJOLLE, Guy; RAGHAVAN, S. (Ed.). *Communications: Wireless in Developing Countries and Networks of the Future*. Springer Boston, 2010, (IFIP Advances in Information and Communication Technology, v. 327). p. 3–14. Disponível em: <http://dx.doi.org/10.1007/978-3-642-15476-8_2>.

Sá, M. P. et al. Middleware de integração de sensores para o desenvolvimento de aplicações móveis sensíveis ao contexto. *II Simpósio Brasileiro de Computação Ubíqua e Pervasiva - XXX Congresso da Sociedade Brasileira de Computação*, p. 221–230, 2010.

SENSEI. Integrating the physical with the digital world of the network of the future. *European 7th framework Programme for Research and Technological Development*, 2010. Disponível em: <<http://www.sensei-project.eu>>.

SOHRABY, K.; MINOLI, D.; ZNATI, T. *Wireless Sensor Networks: technology, protocols and applications*. [S.l.]: Willey-Interscience, 2007.

TINYOS. *An open-source OS for the networked sensor regime*. 2010. Disponível em: <<http://www.tinyos.net>>.

TINYOS. *TOSSIM*. 2010. Disponível em: <<http://docs.tinyos.net/index.php/TOSSIM>>.

XIA, F; ZHAO, W.; TIAN, Y. C. Fuzzy control based qos management in wireless sensor/actuator networks. *Sensors 2007*, 2007. Disponível em: <www.mdpi.org/sensors>.